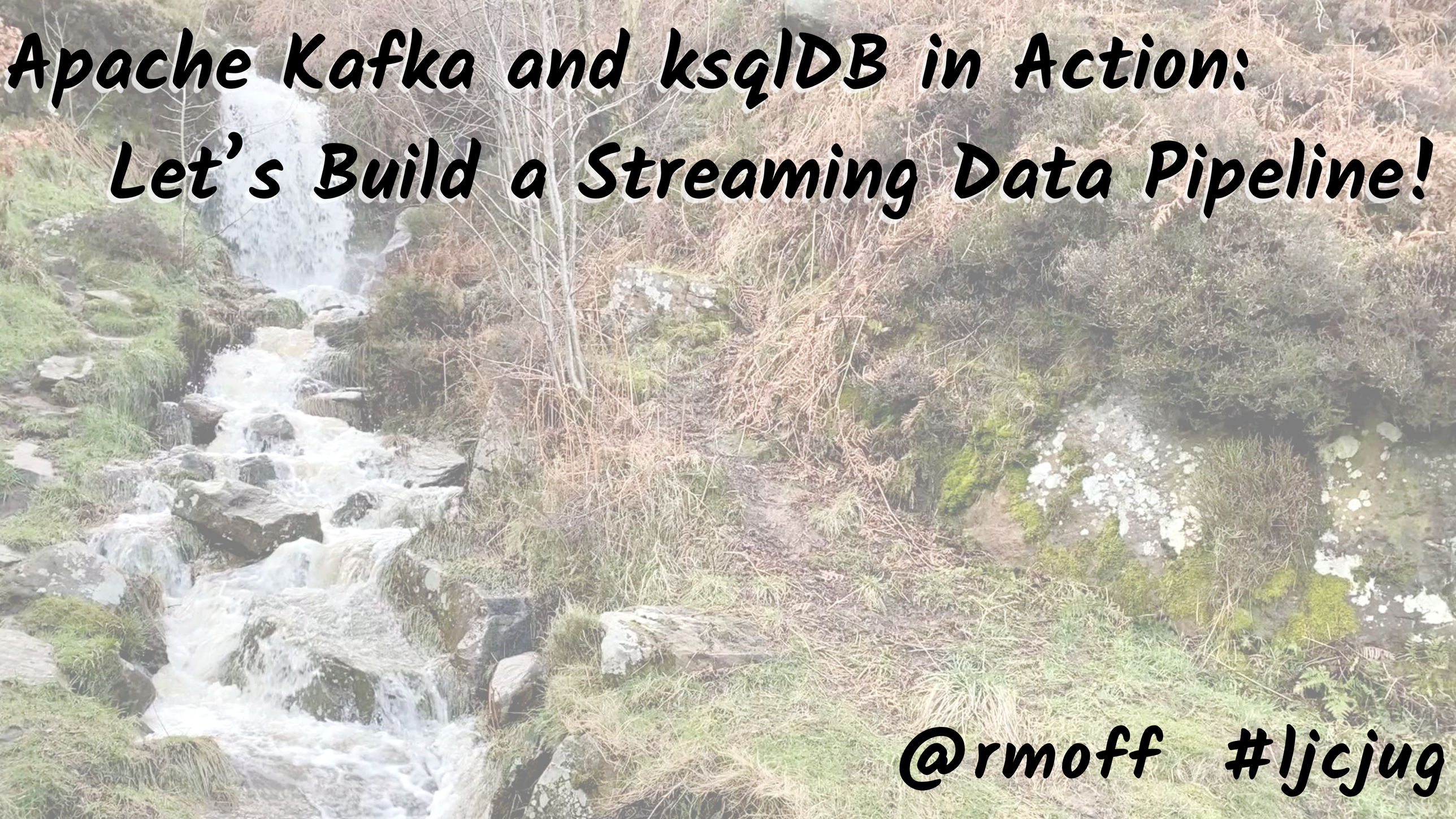
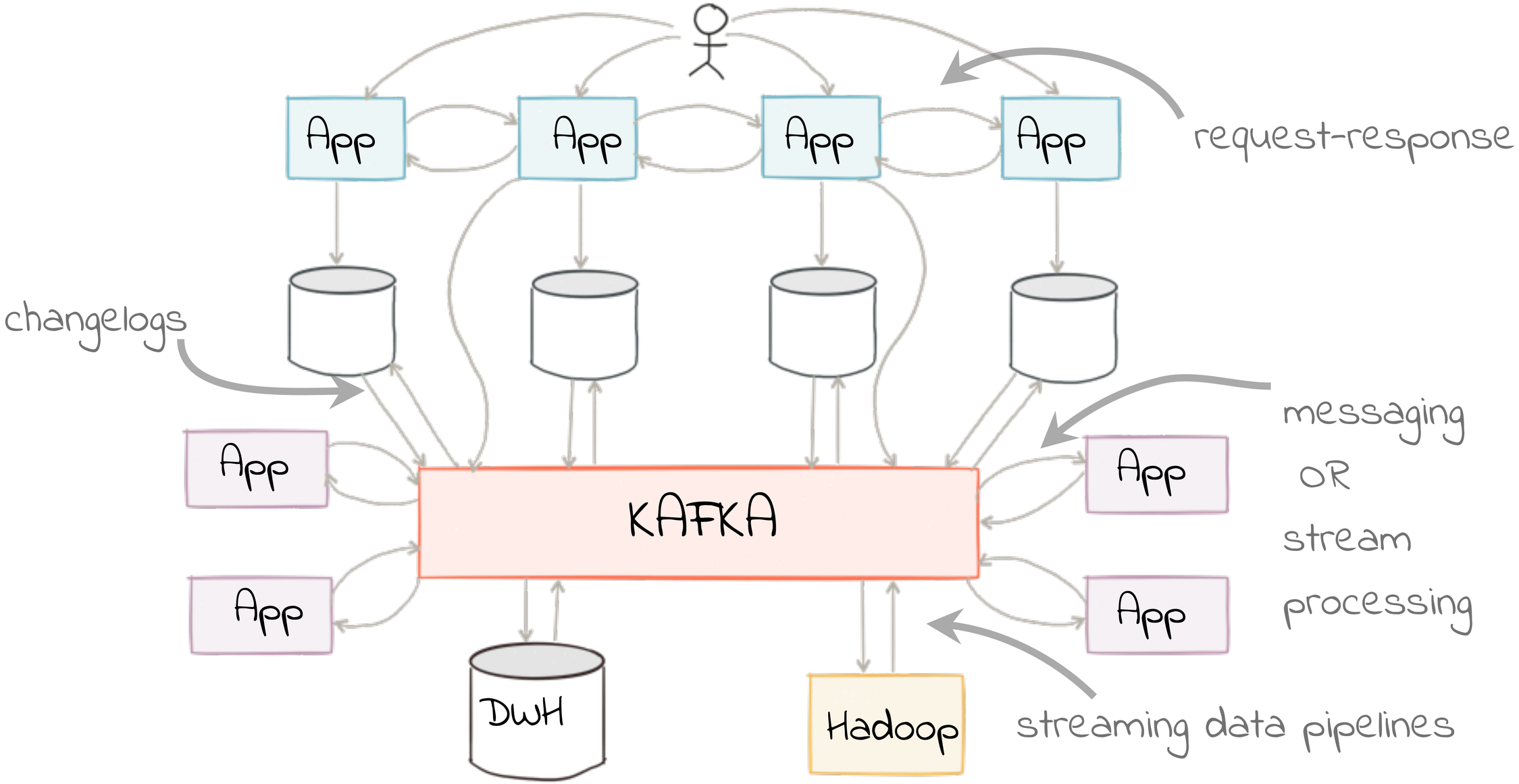
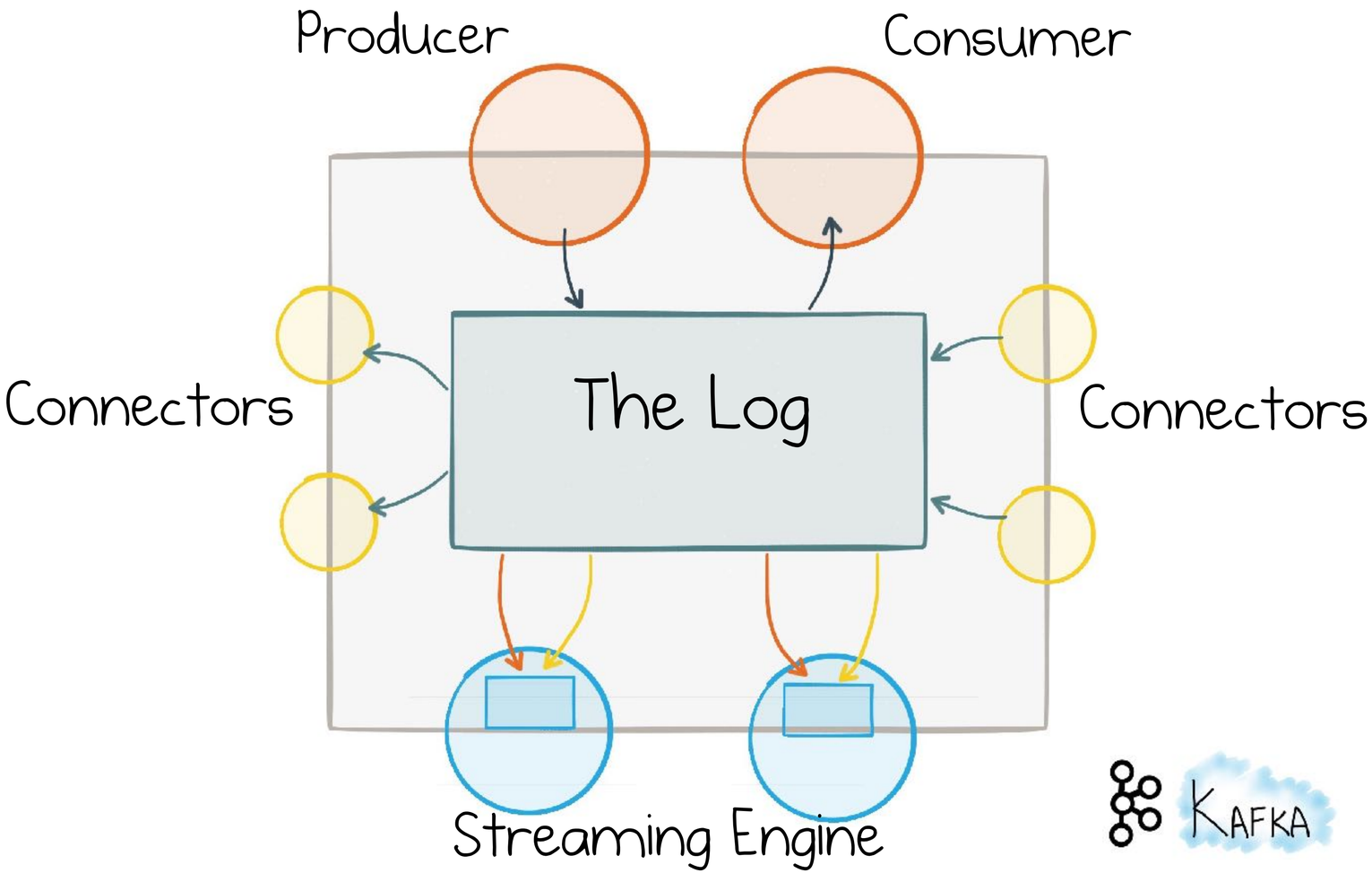

Apache Kafka and ksqlDB in Action:
Let's Build a Streaming Data Pipeline!
@rmoff #ljcjug

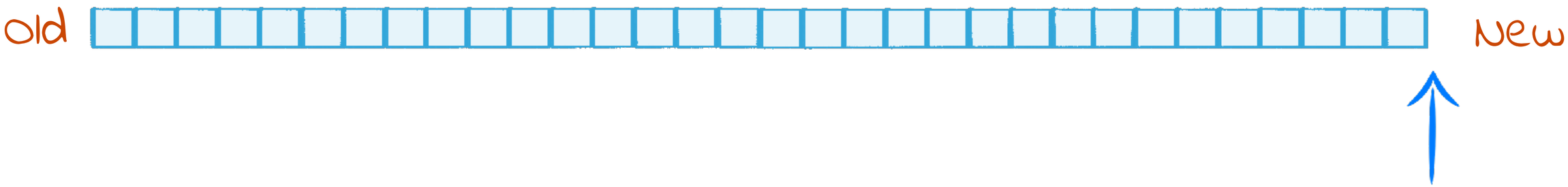
Kafka is an Event Streaming Platform



What is an Event Streaming Platform?

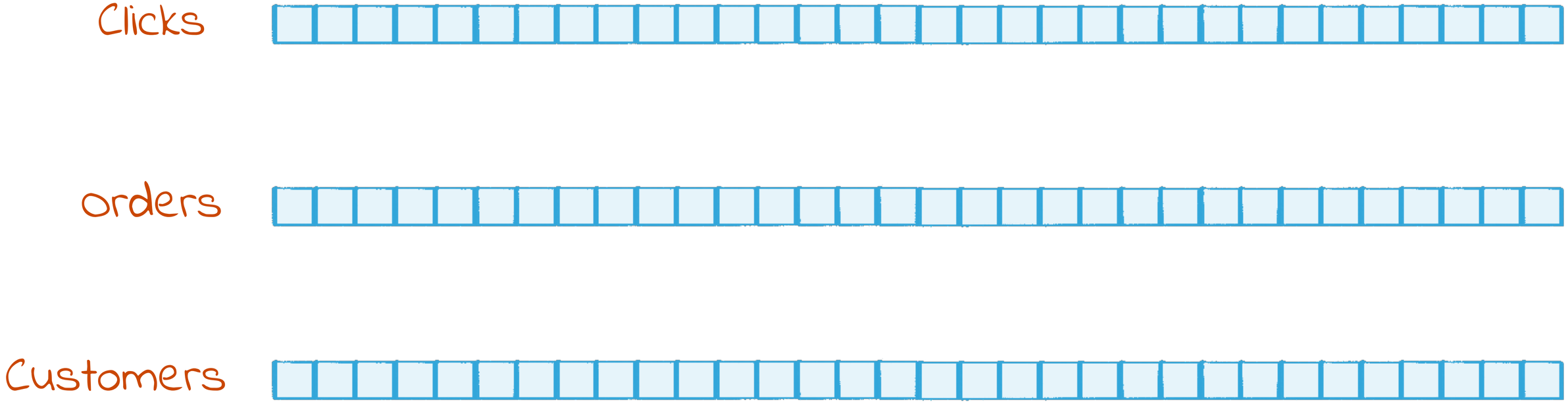


Immutable Event Log



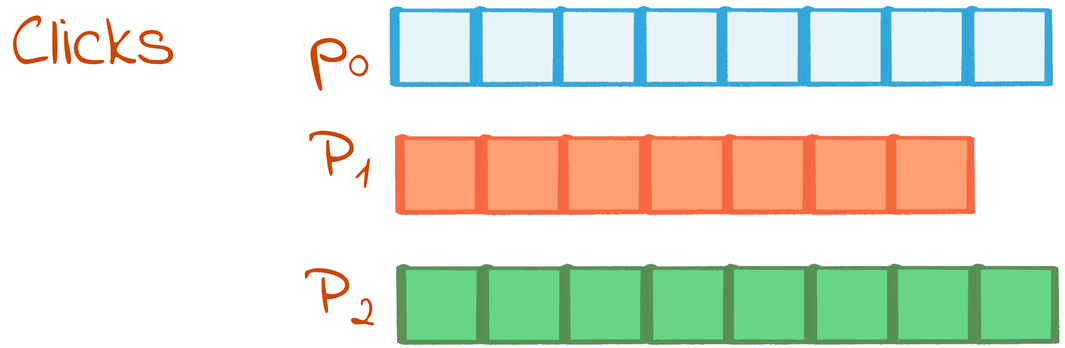
Messages are added at the end of the log

Topics



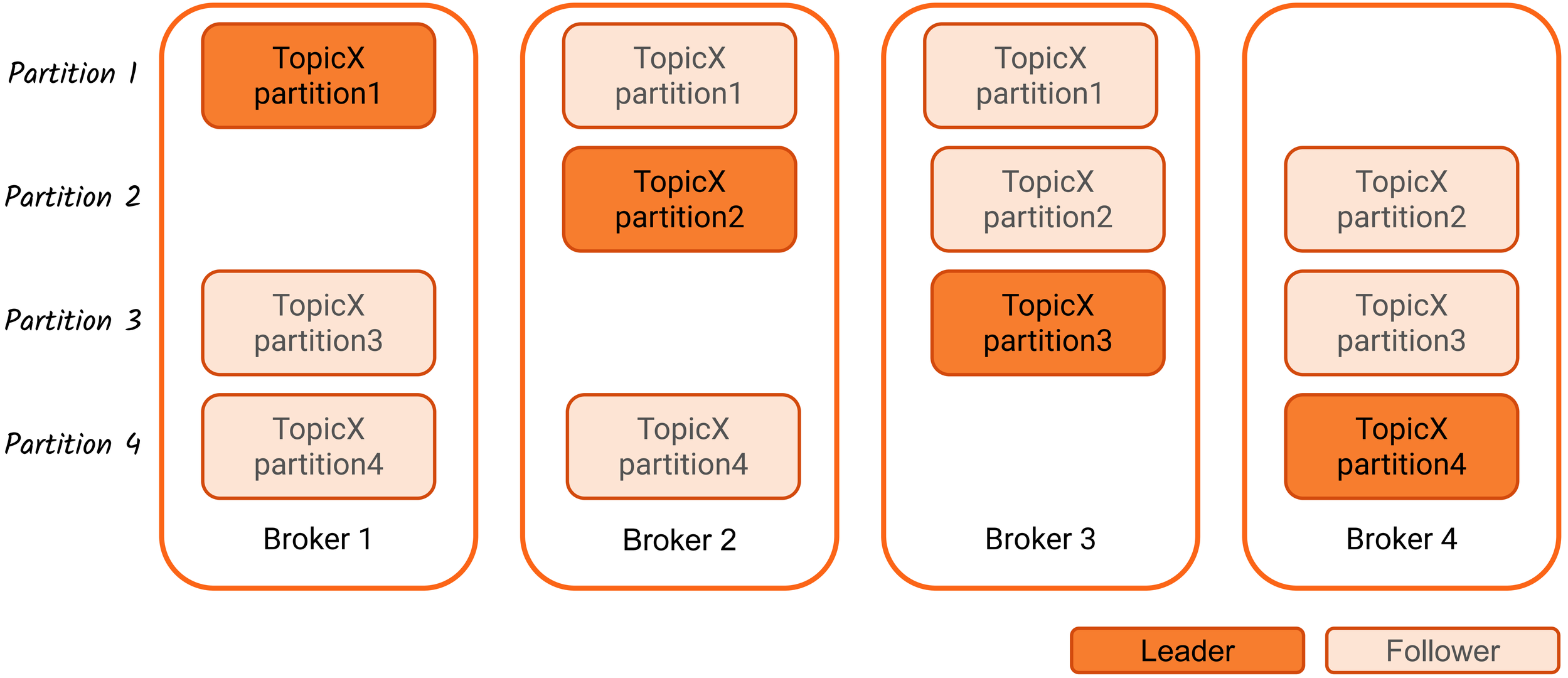
Topics are similar in concept to tables in a database

Partitions

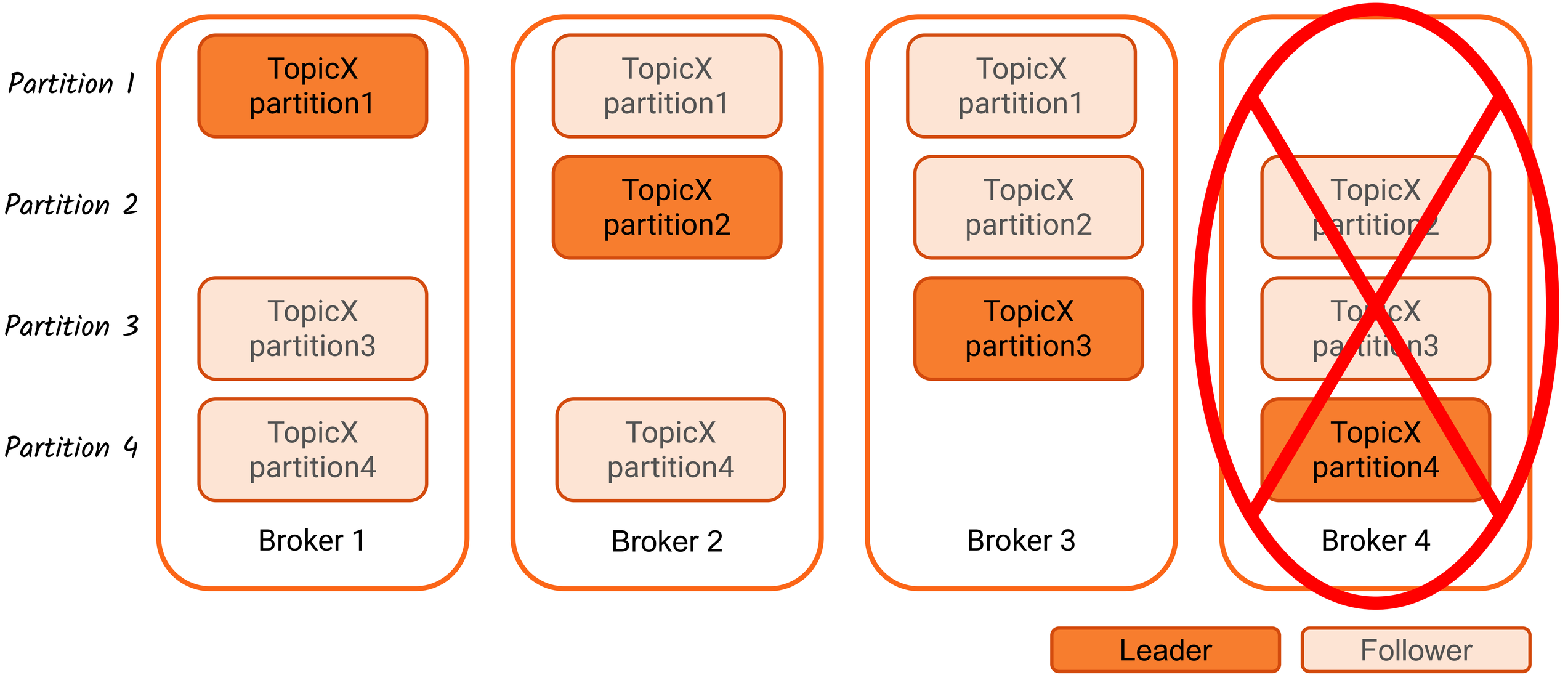


Messages are guaranteed to be strictly ordered within a partition

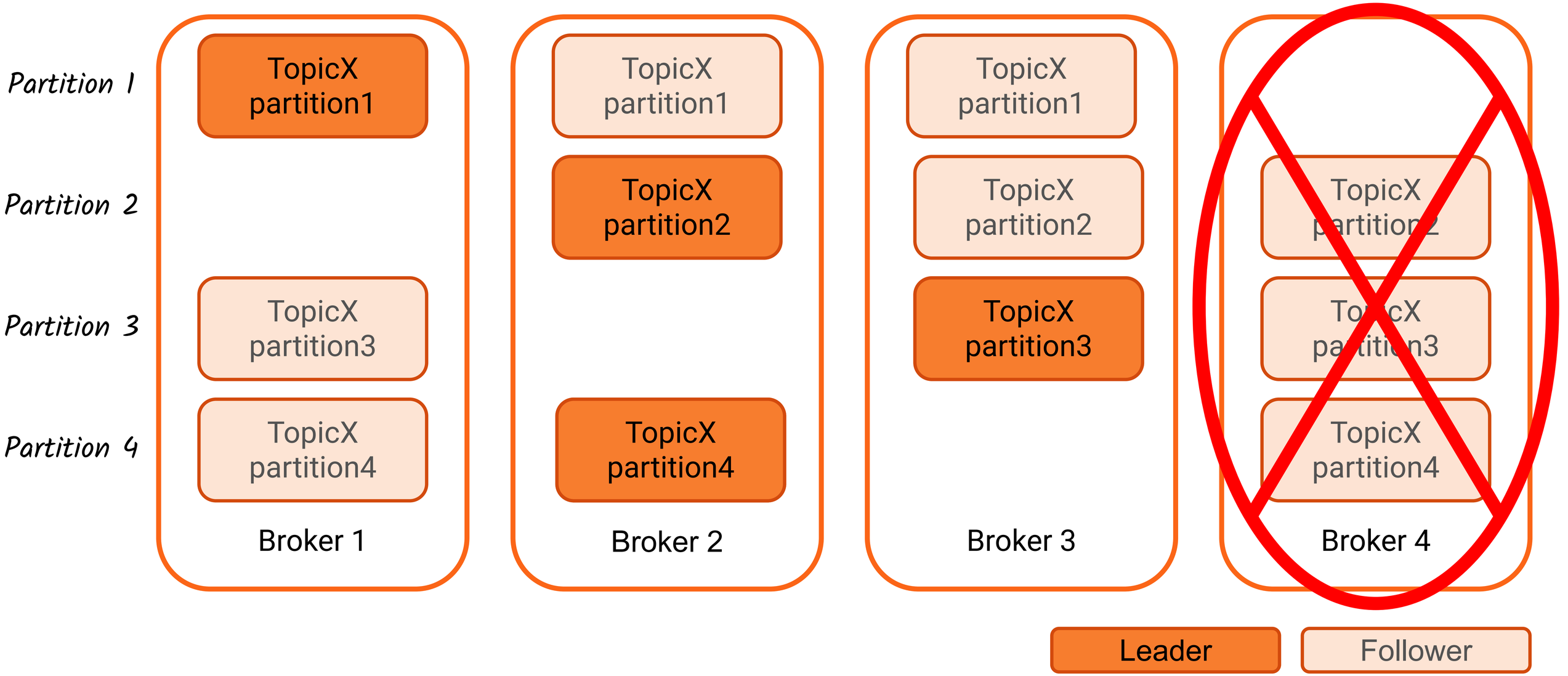
Partition Leadership and Replication



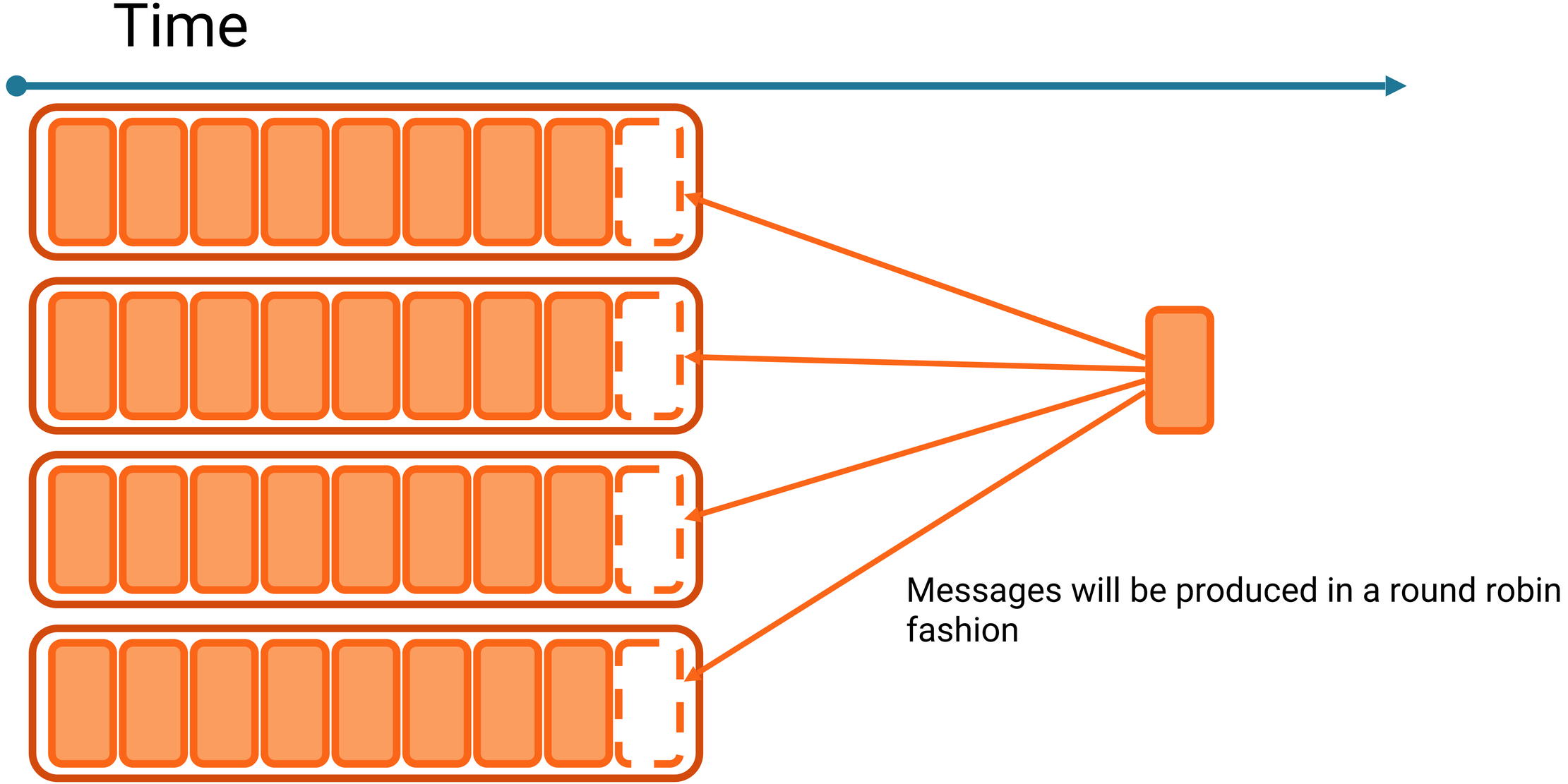
Partition Leadership and Replication



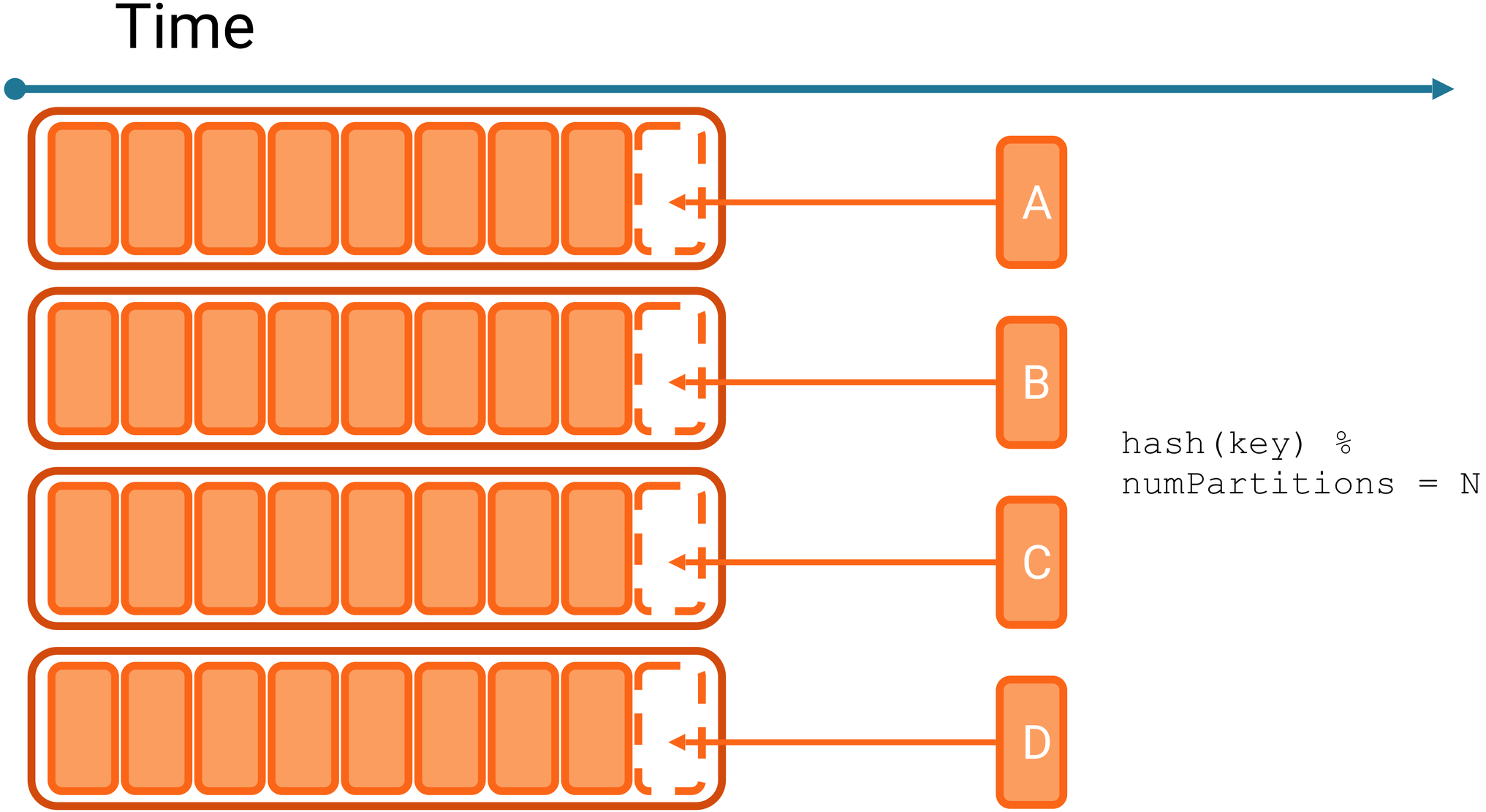
Partition Leadership and Replication



Producing to Kafka - No Key

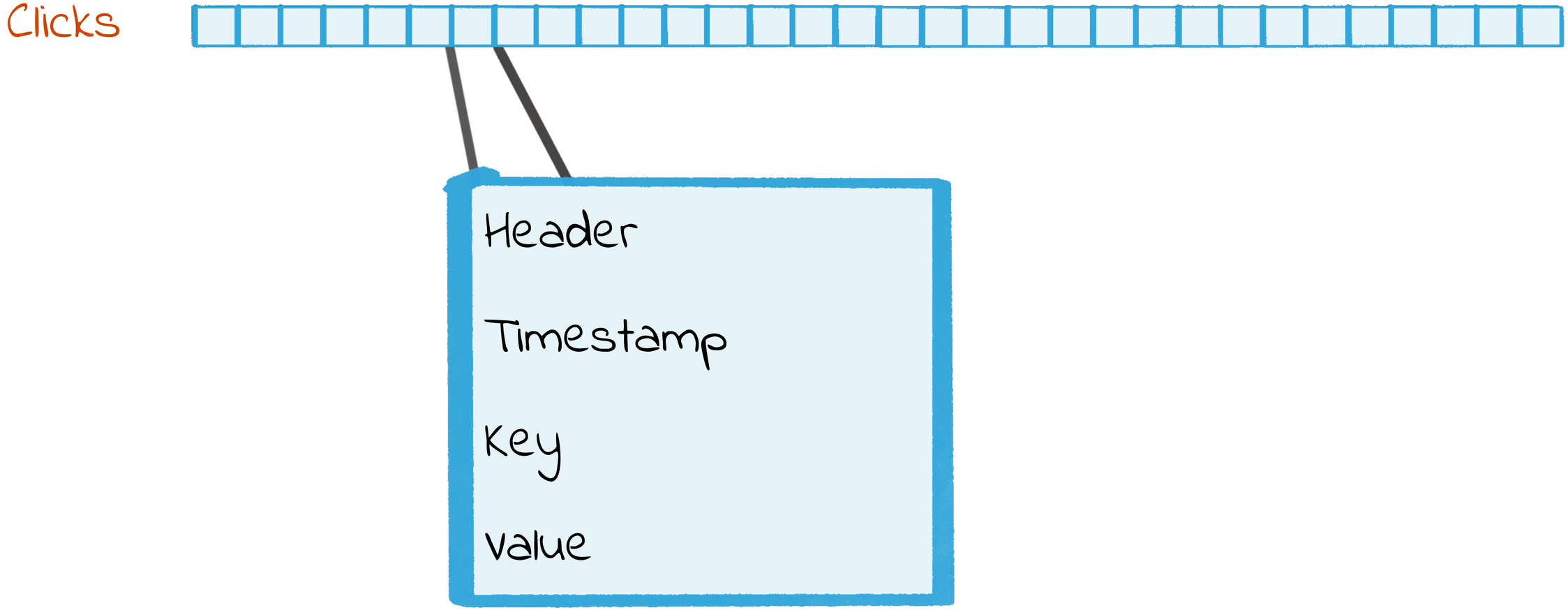


Producing to Kafka - With Key



Messages are just KN bytes

plus headers + timestamp



Messages are just KN bytes

With great power comes great responsibility

Avro

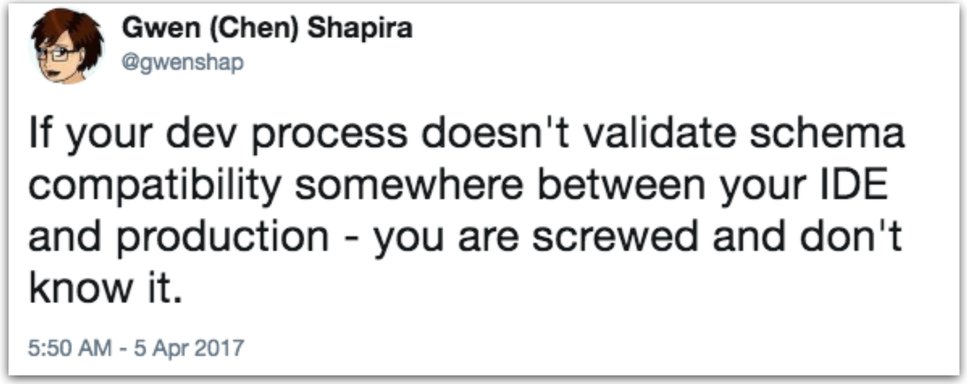
-> Confluent Schema Registry

Protobuf

-> Confluent Schema Registry

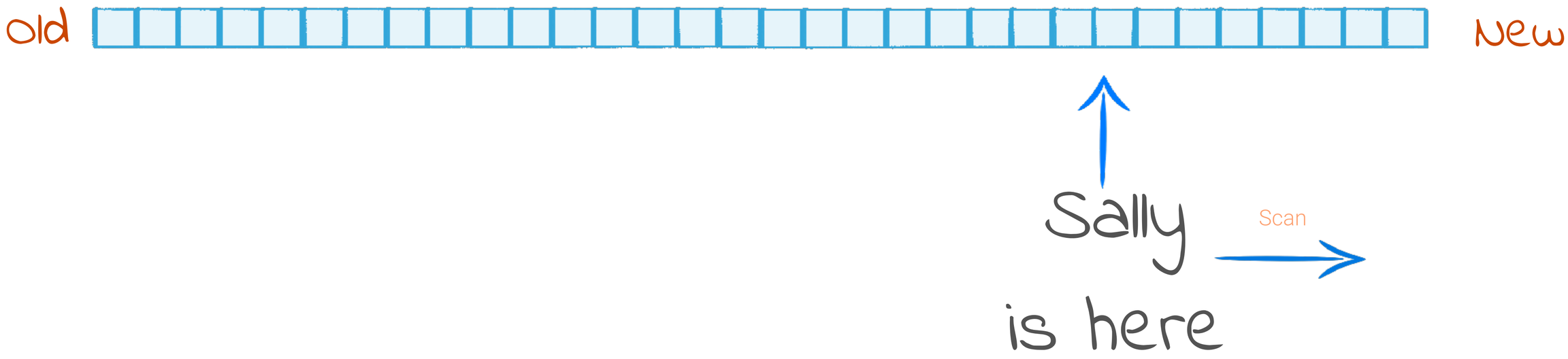
JSON

CSV

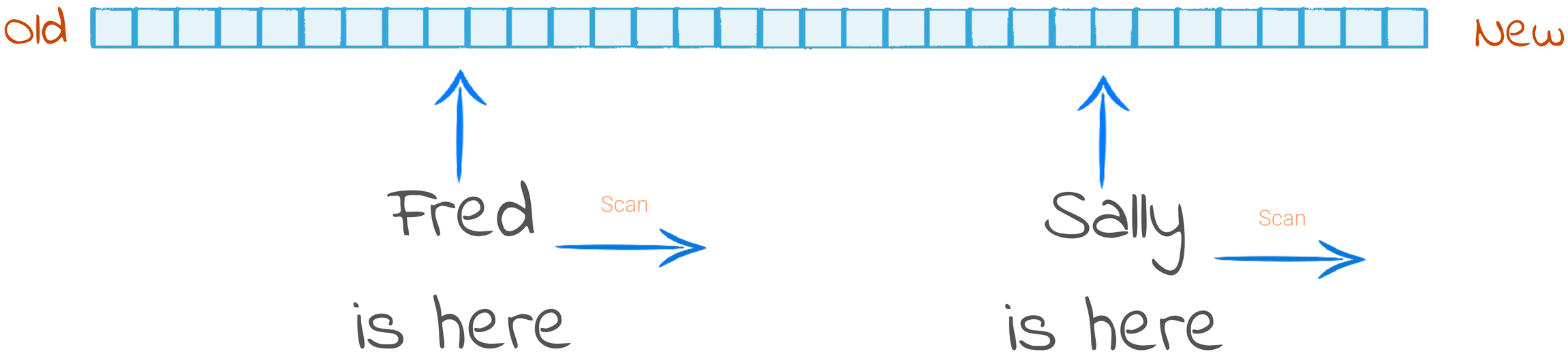


https://qconnewyork.com/system/files/presentation-slides/qcon_17_-_schemas_and_apis.pdf

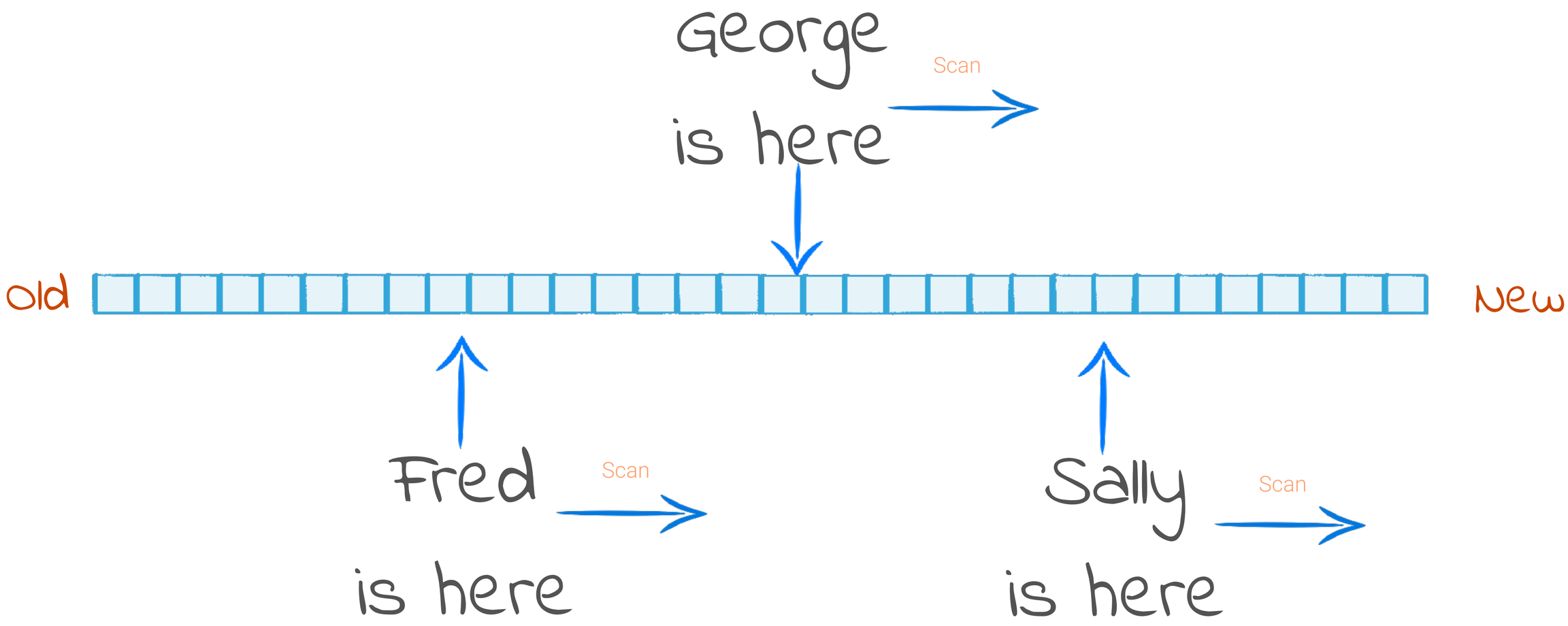
Consumers have a position all of their own



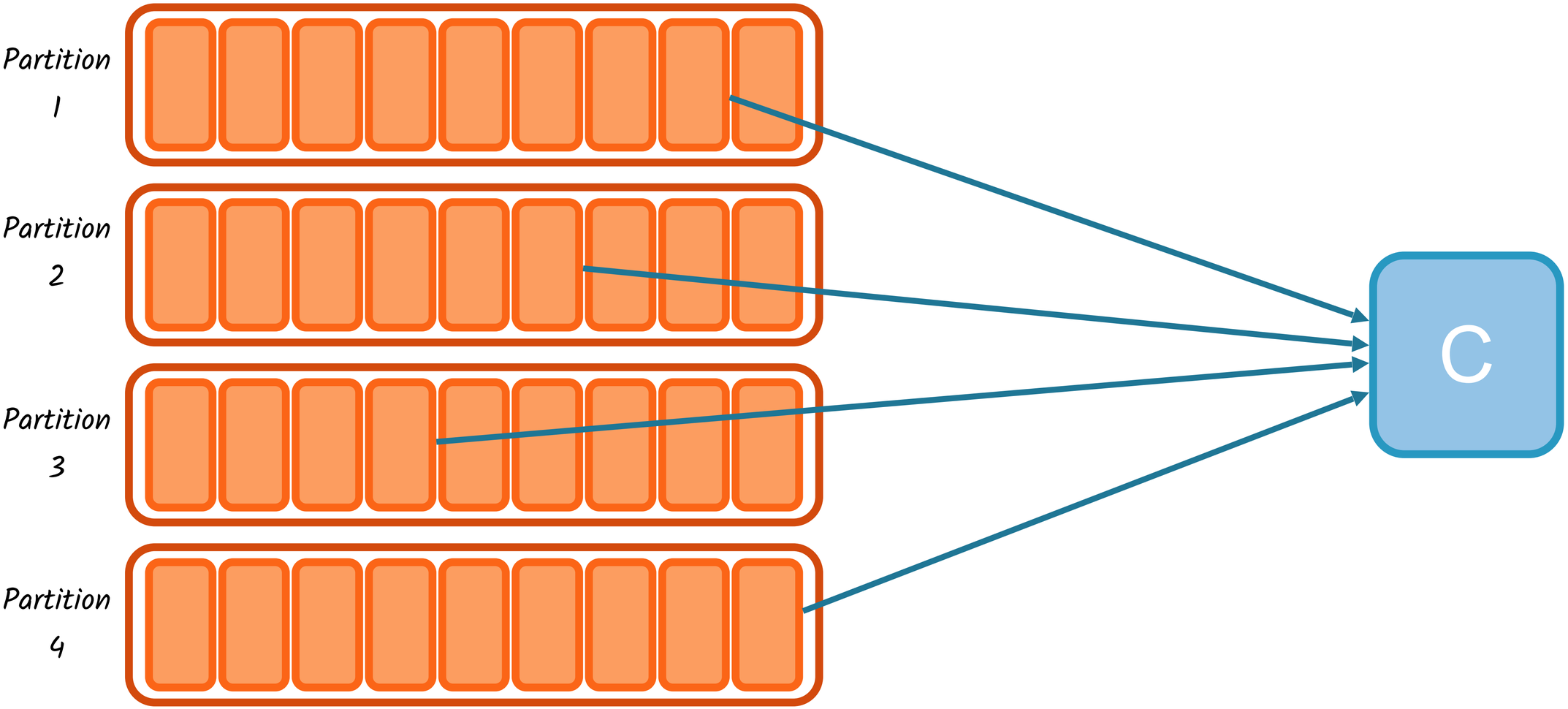
Consumers have a position all of their own



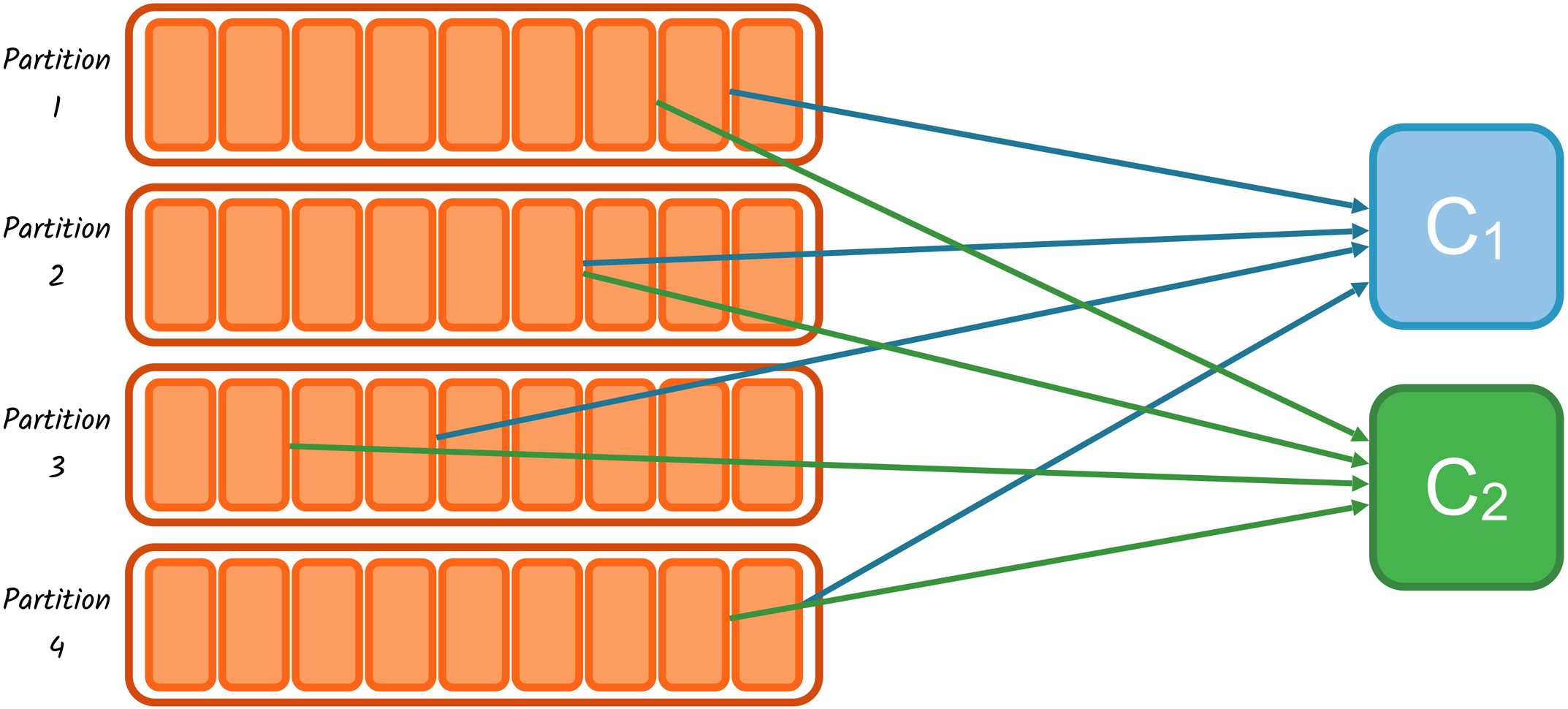
Consumers have a position all of their own



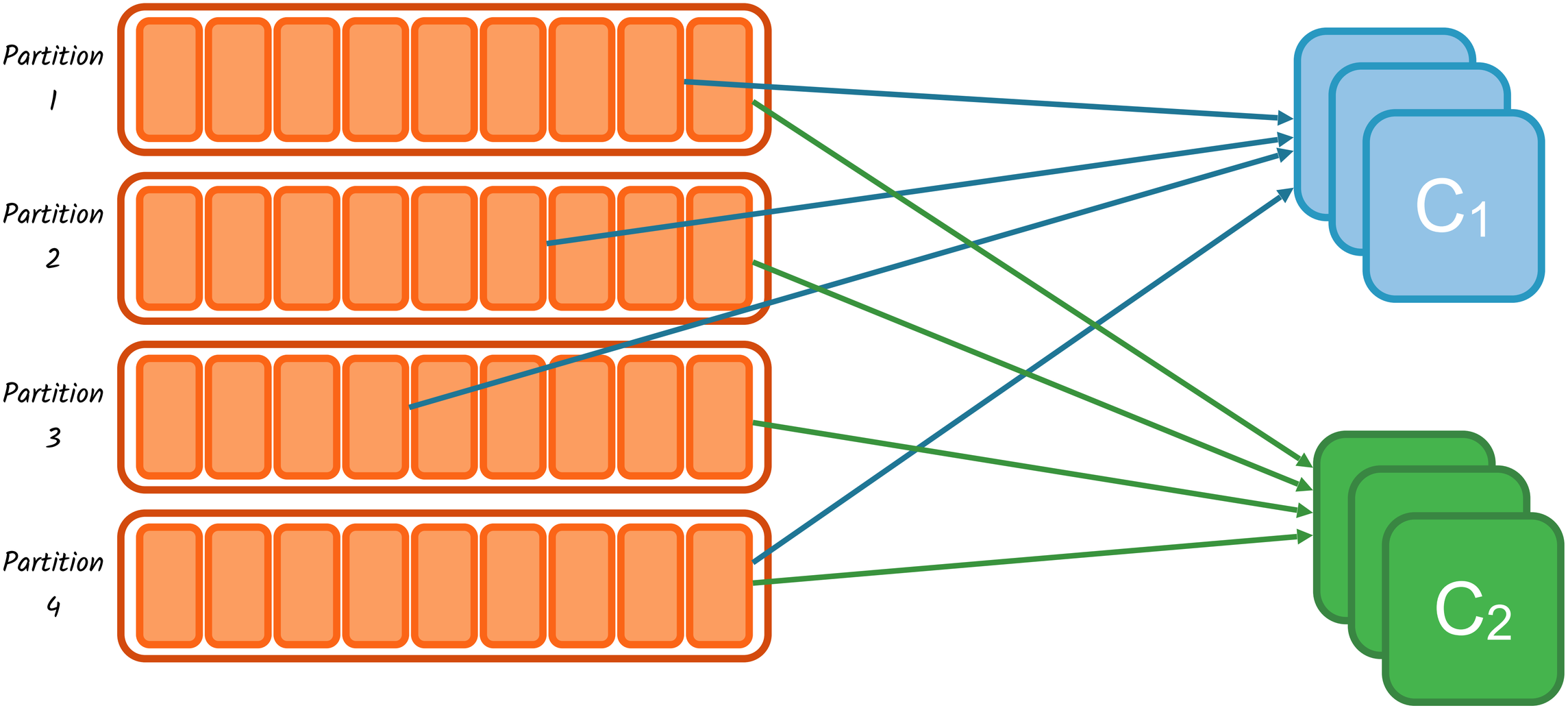
Consuming From Kafka - Single Consumer



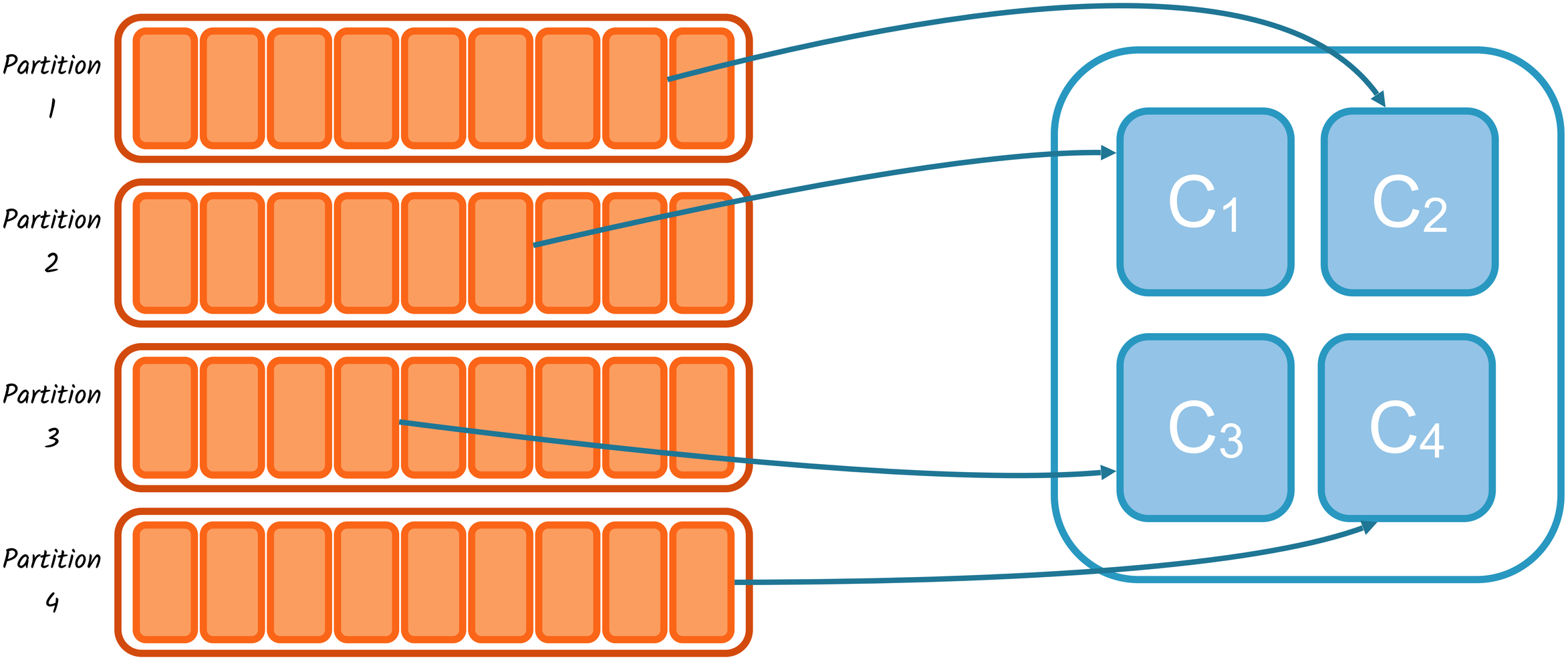
Consuming From Kafka - Multiple Consumers



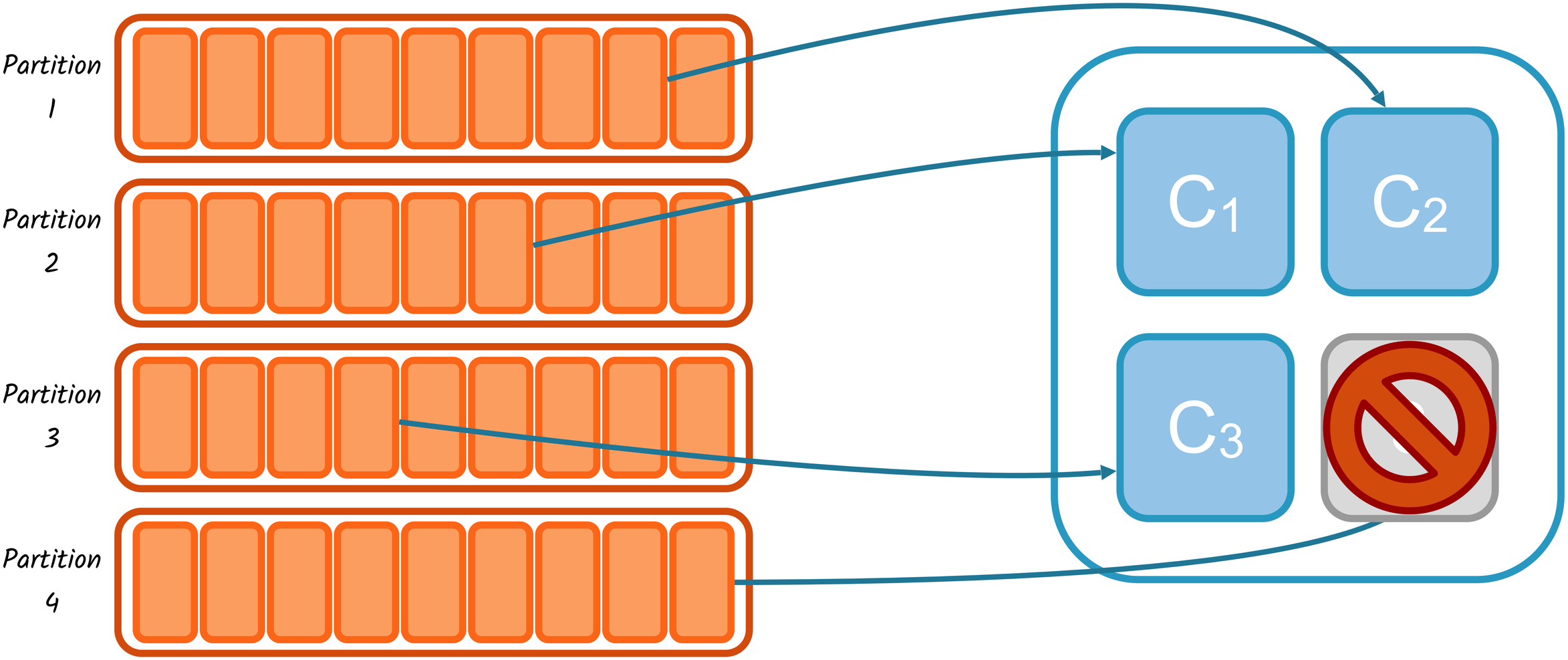
Consuming From Kafka - Grouped Consumers



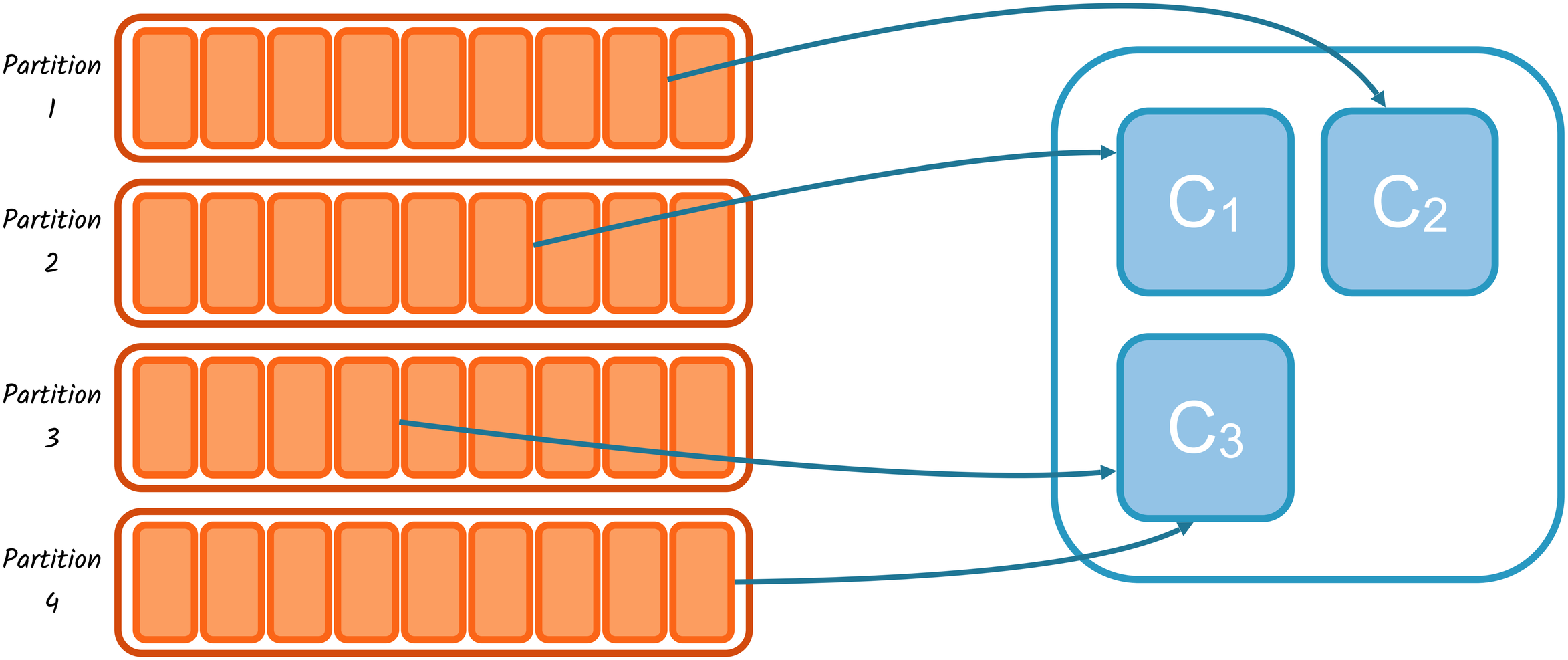
Consuming From Kafka - Grouped Consumers



Consuming From Kafka - Grouped Consumers

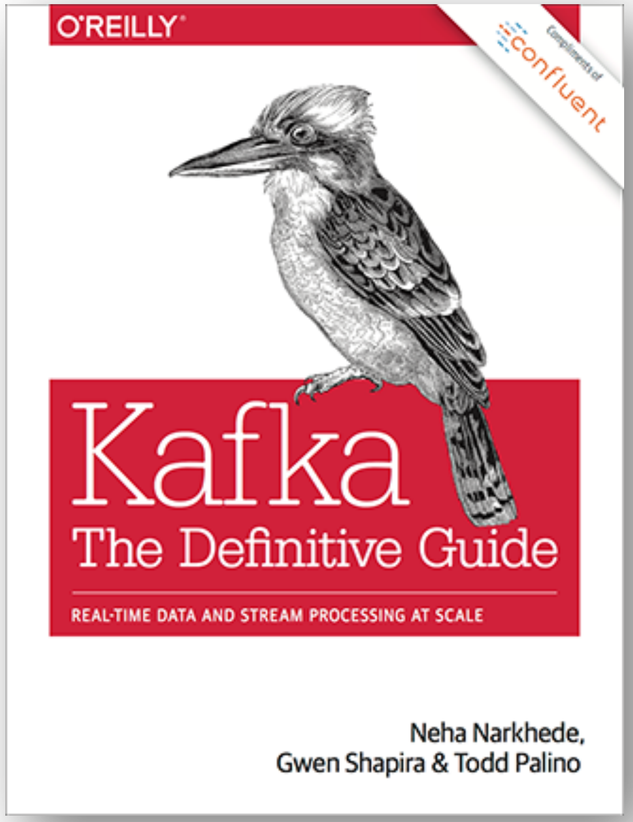


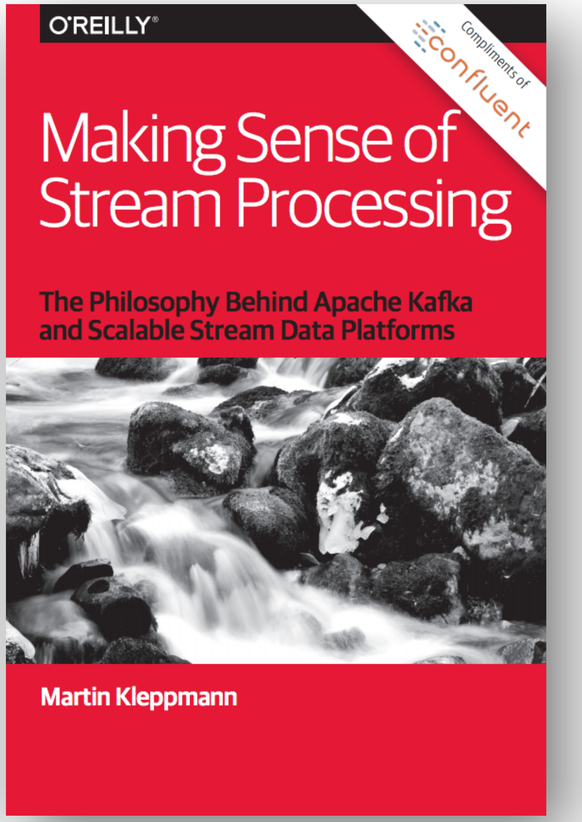
Consuming From Kafka - Grouped Consumers

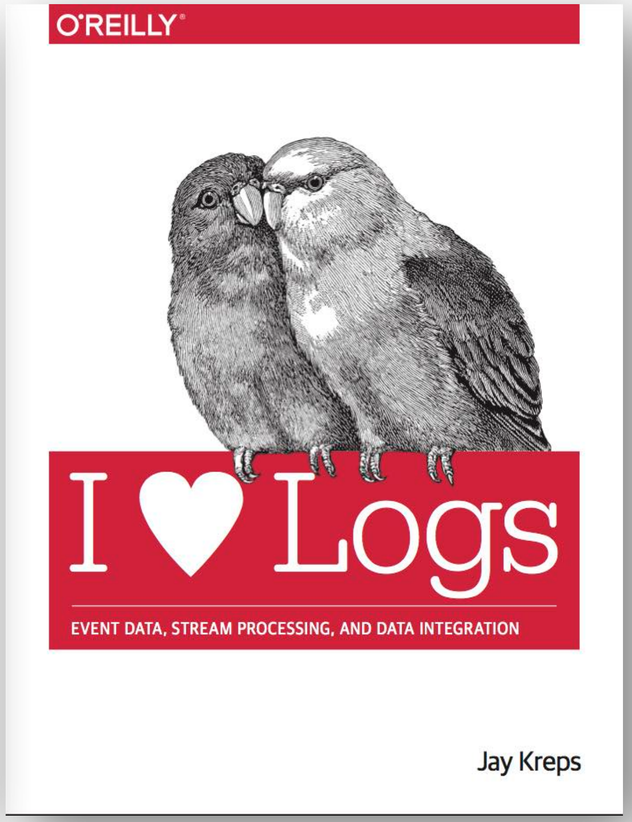


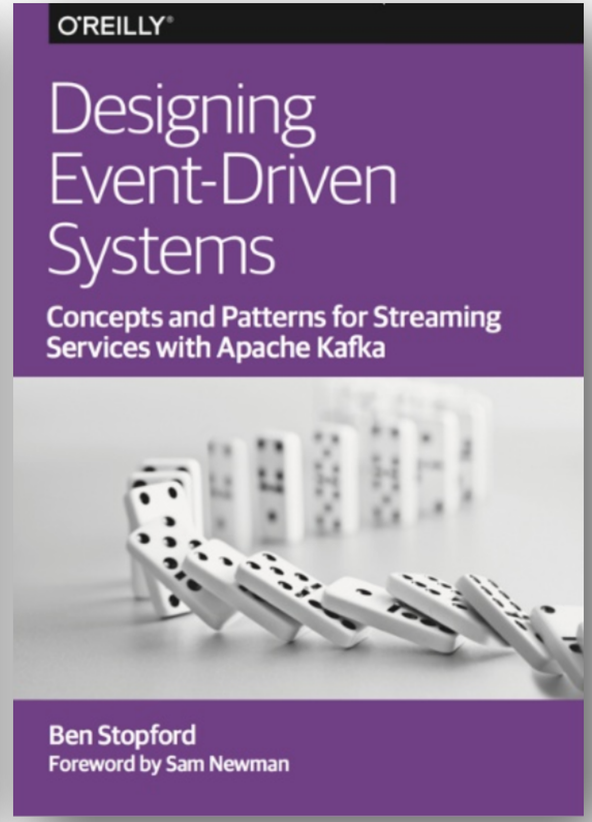
Free Books!

https://rmoff.dev/ljcjug









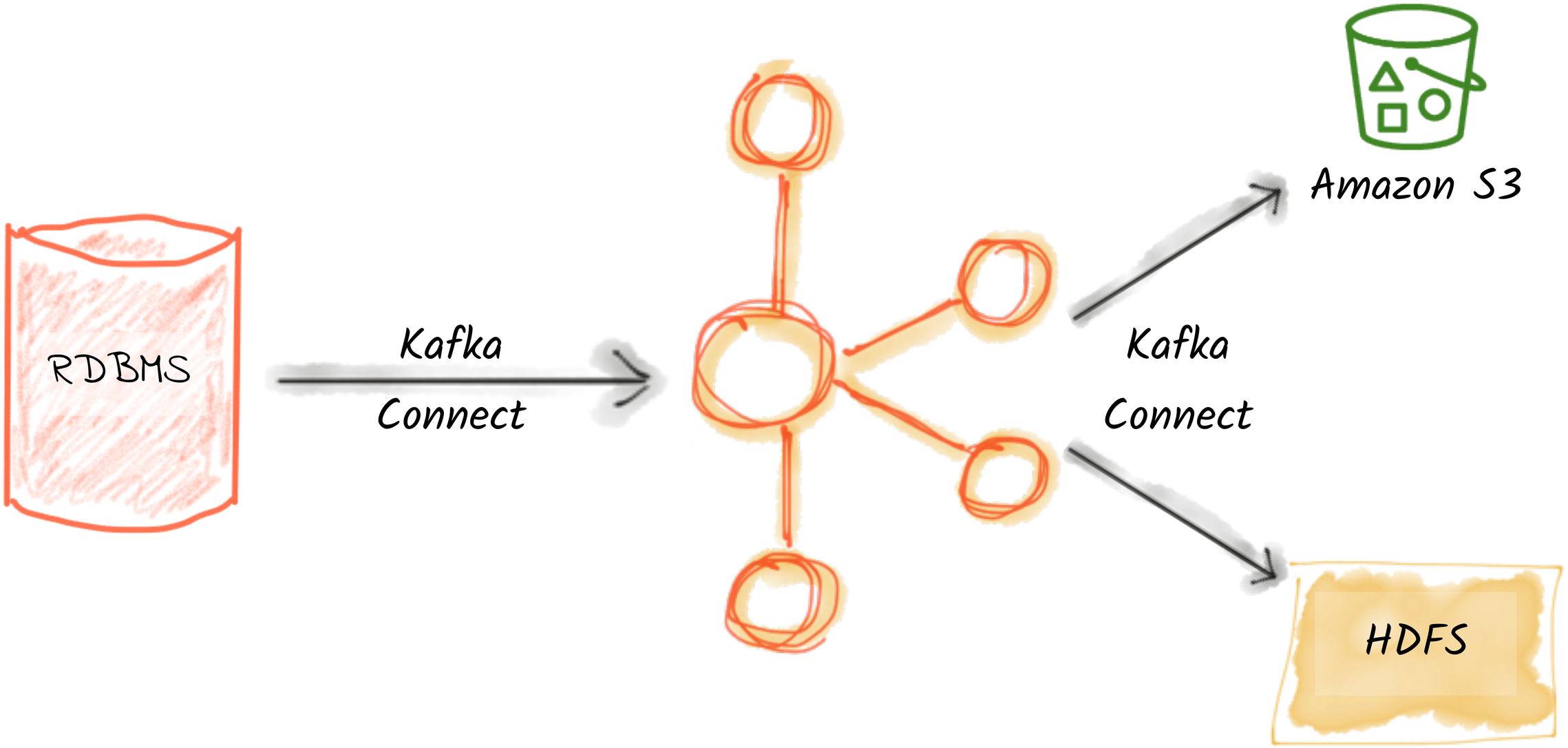
@rmoff #ljcjug

Streaming Data Pipelines

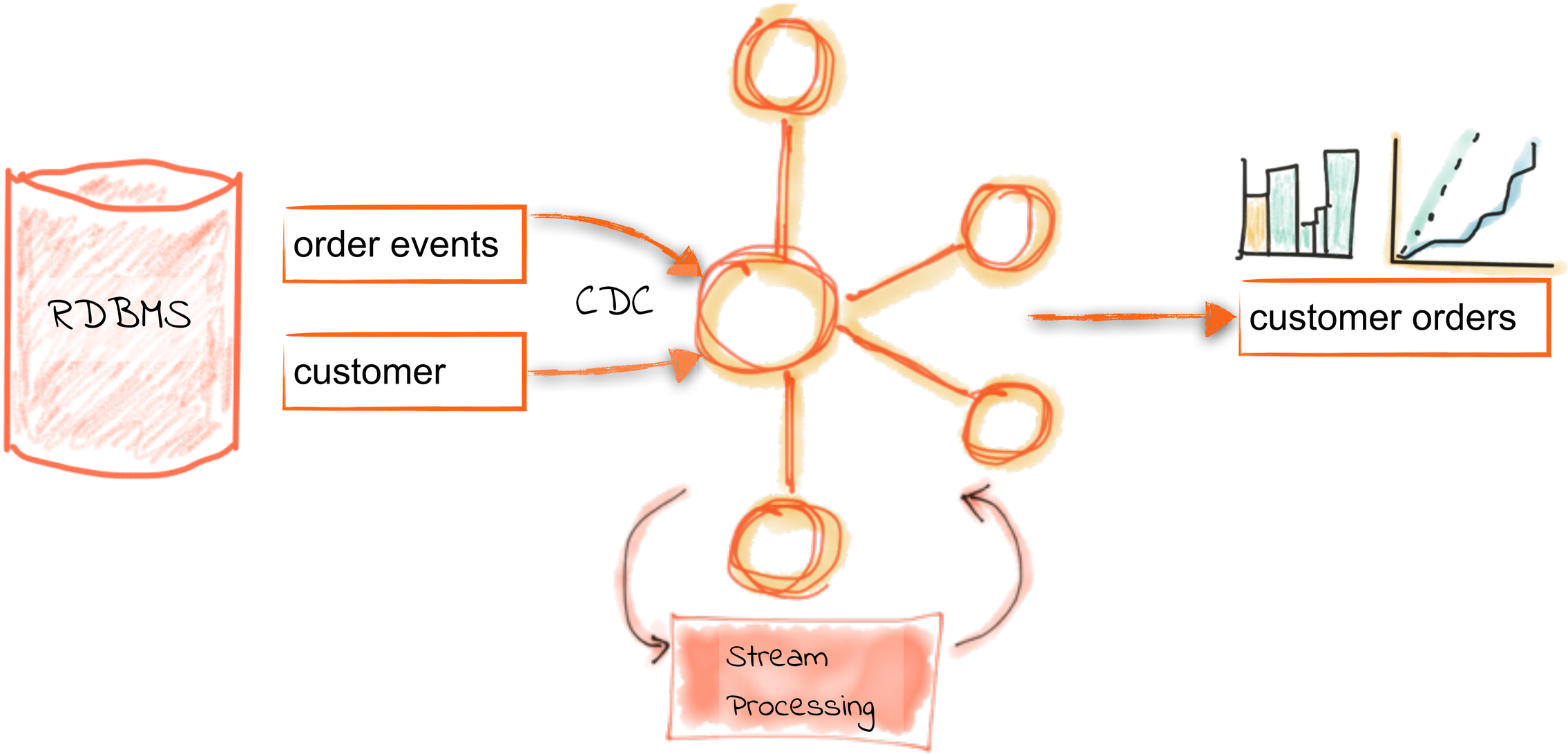
confluent

Photo by Victor Garcia on Unsplash

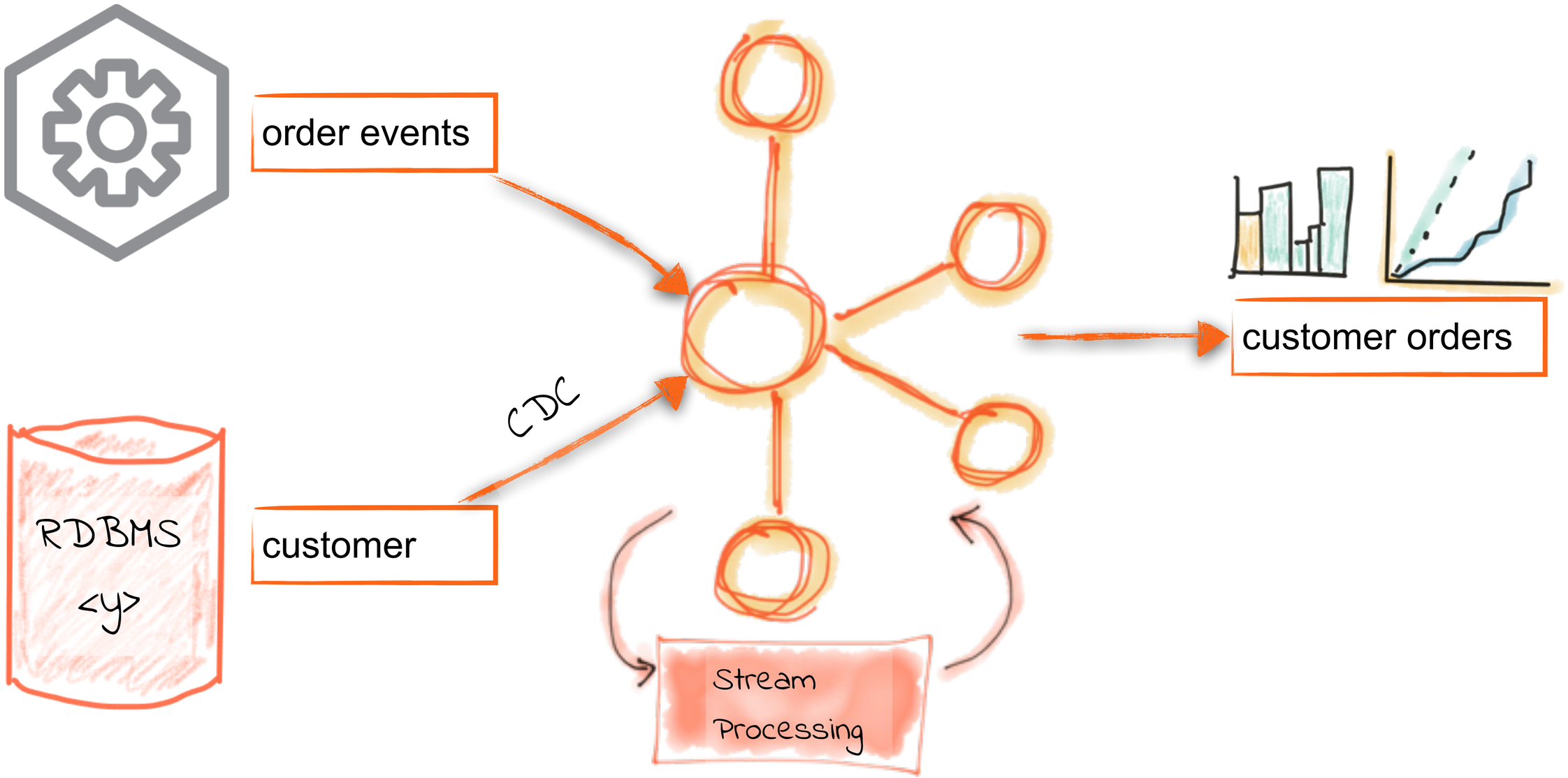
Database Offload



Stream Processing with Apache Kafka and ksqlDB



Real-time Event Stream Enrichment



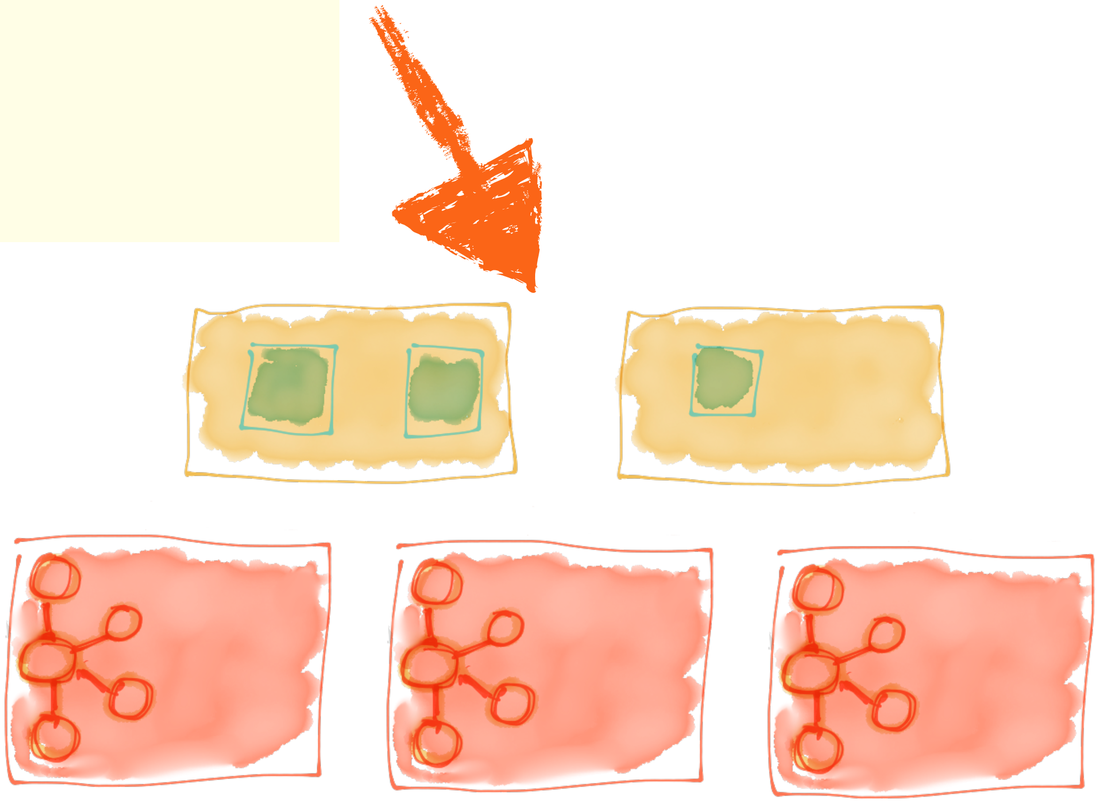
Building a streaming data pipeline

Integration + Stream Processing

Integration

Streaming Integration with Kafka Connect

Sources

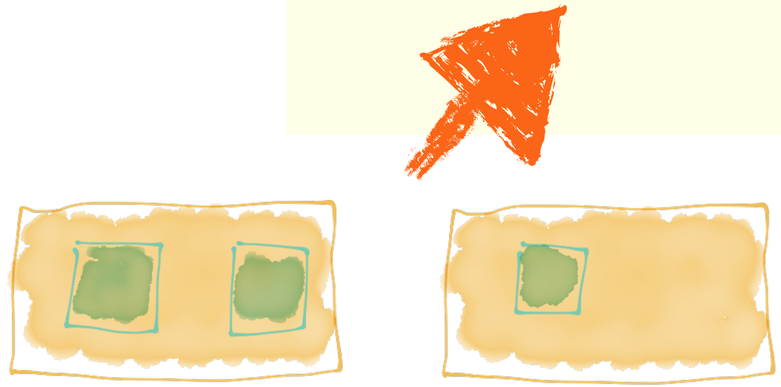


Kafka Connect

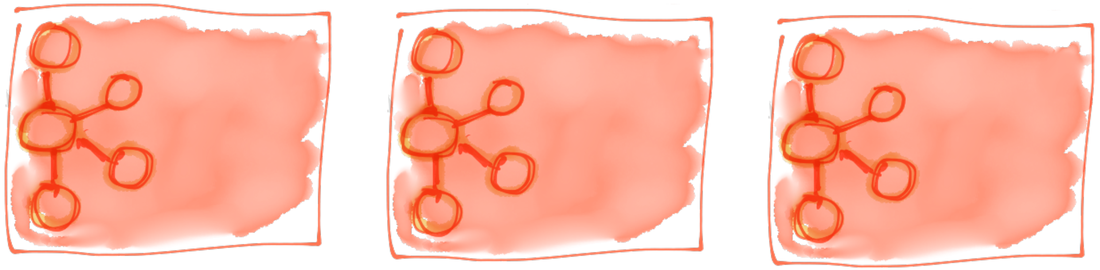
Kafka Brokers

Streaming Integration with Kafka Connect

Sinks

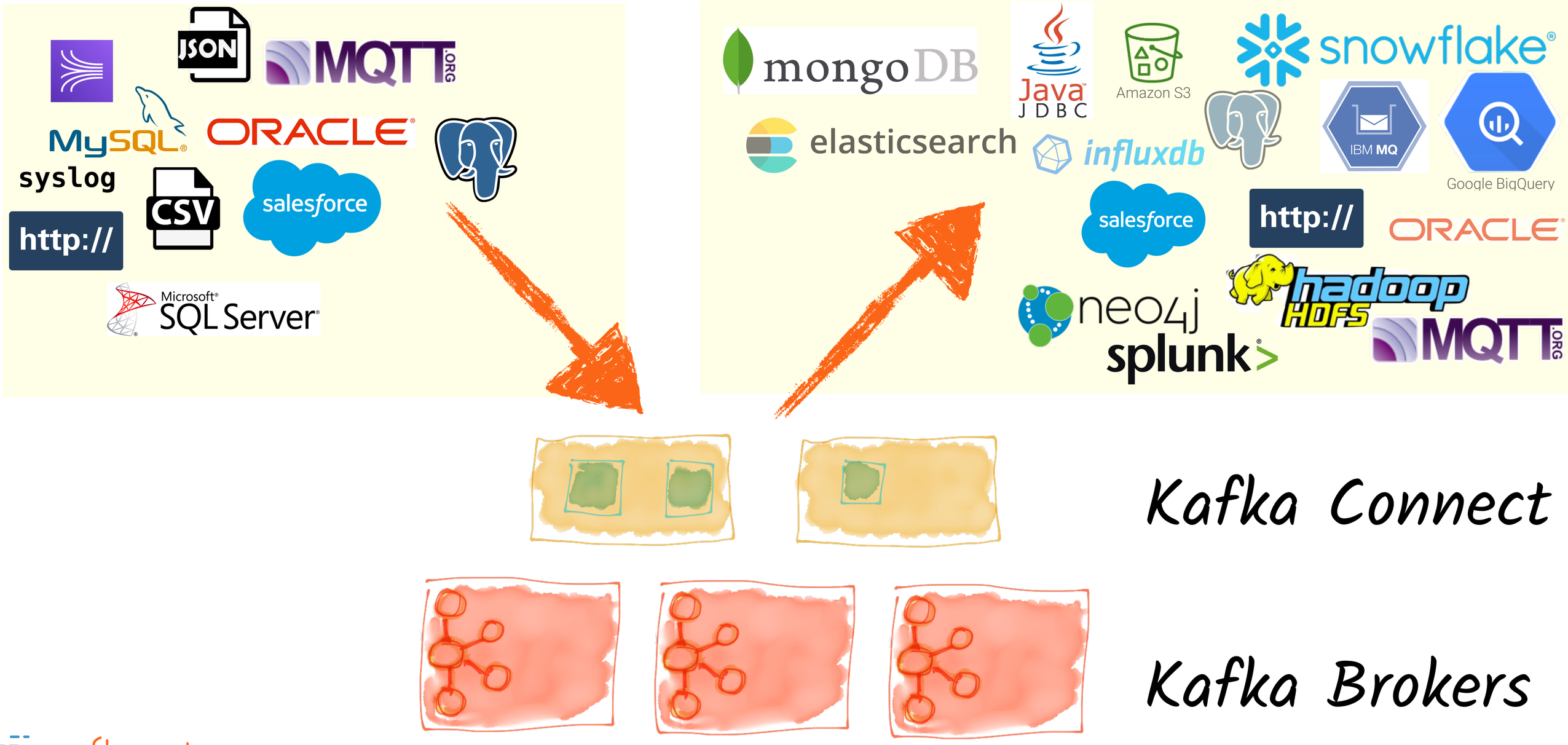


Kafka Connect



Kafka Brokers

Streaming Integration with Kafka Connect



Look Ma, No Code!

```
{
  "connector.class":
    "io.confluent.connect.jdbc.JdbcSourceConnector",
  "connection.url":
    "jdbc:mysql://asgard:3306/demo",
  "table.whitelist":
    "sales,orders,customers"
}
```

https://docs.confluent.io/current/connect/

Messages are just KN bytes

With great power comes great responsibility

Avro

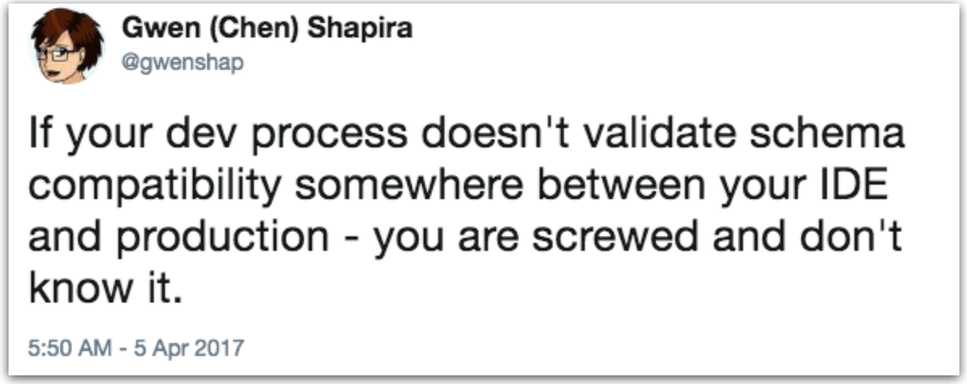
-> Confluent Schema Registry

Protobuf

-> Confluent Schema Registry

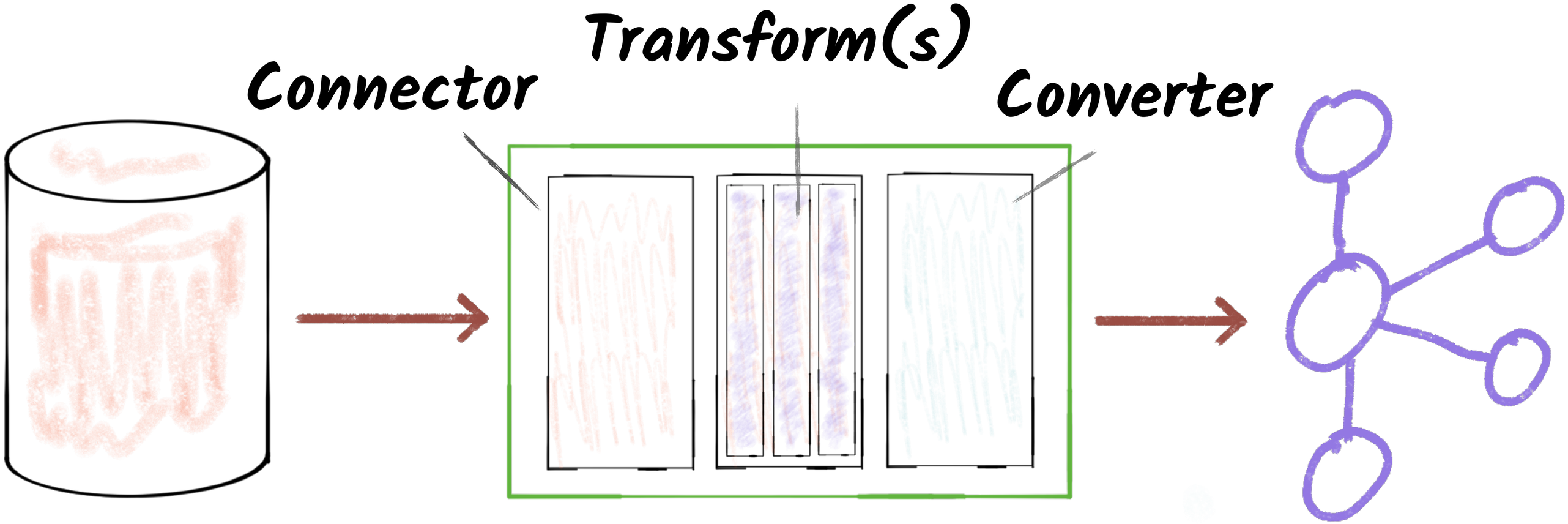
JSON

CSV

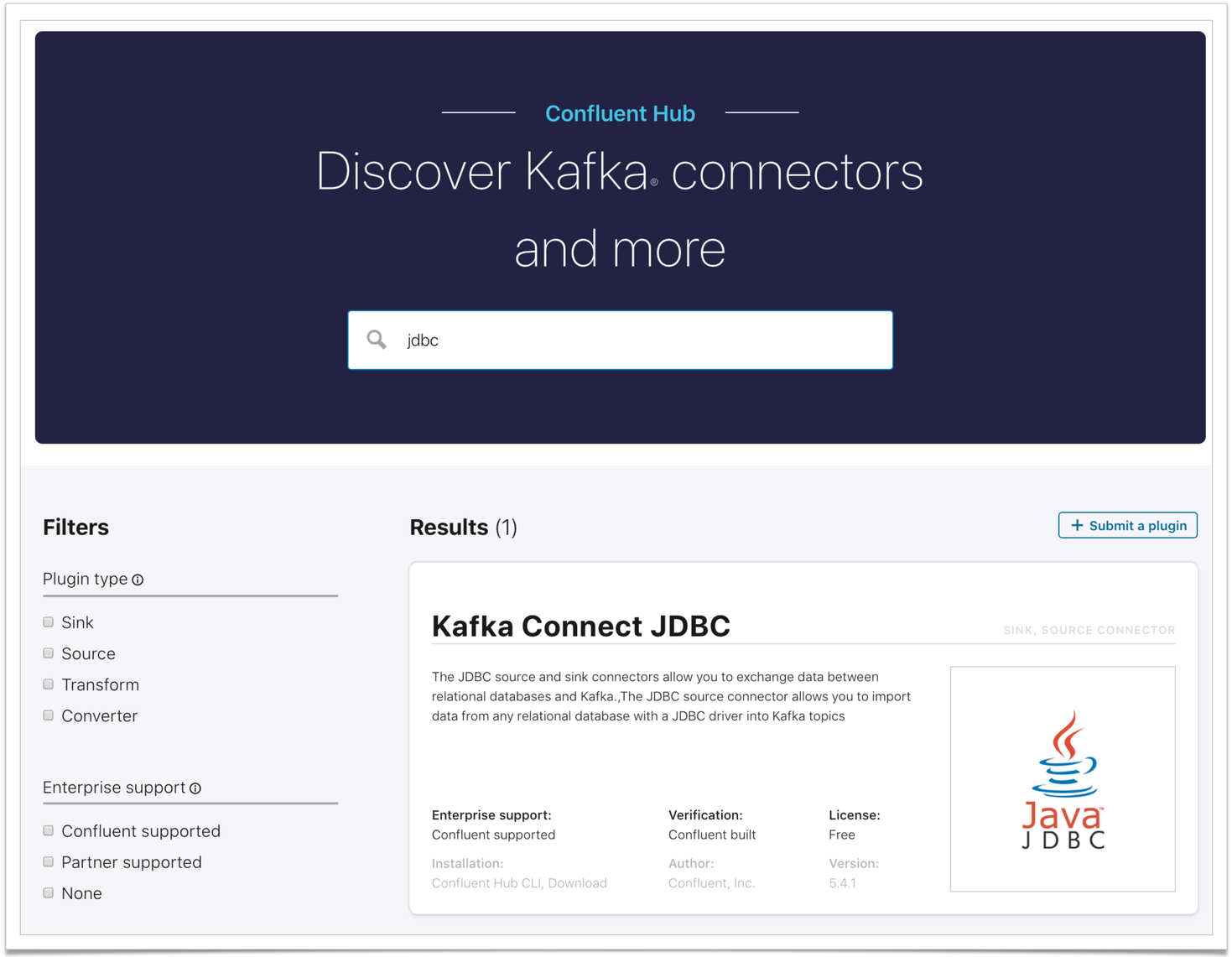


https://qconnewyork.com/system/files/presentation-slides/qcon_17_-_schemas_and_apis.pdf

Extensible



Confluent Hub



hub.confluent.io

Integration + Stream Processing

Stream Processing

Photo by Franck V. on Unsplash

```
SELECT *
FROM WIDGETS
WHERE WEIGHT_G > 120
```

```
SELECT COUNT(*)
FROM WIDGETS
GROUP BY PRODUCTION_LINE
```

```
SELECT AVG(TEMP_CELCIUS) AS TEMP
FROM WIDGETS
GROUP BY SENSOR_ID
HAVING TEMP>20
```

Object store, data warehouse, RDBMS

```
CREATE SINK CONNECTOR dw WITH (
'connector.class' = 'S3Connector',
'topics' = 'widgets'
…);
```

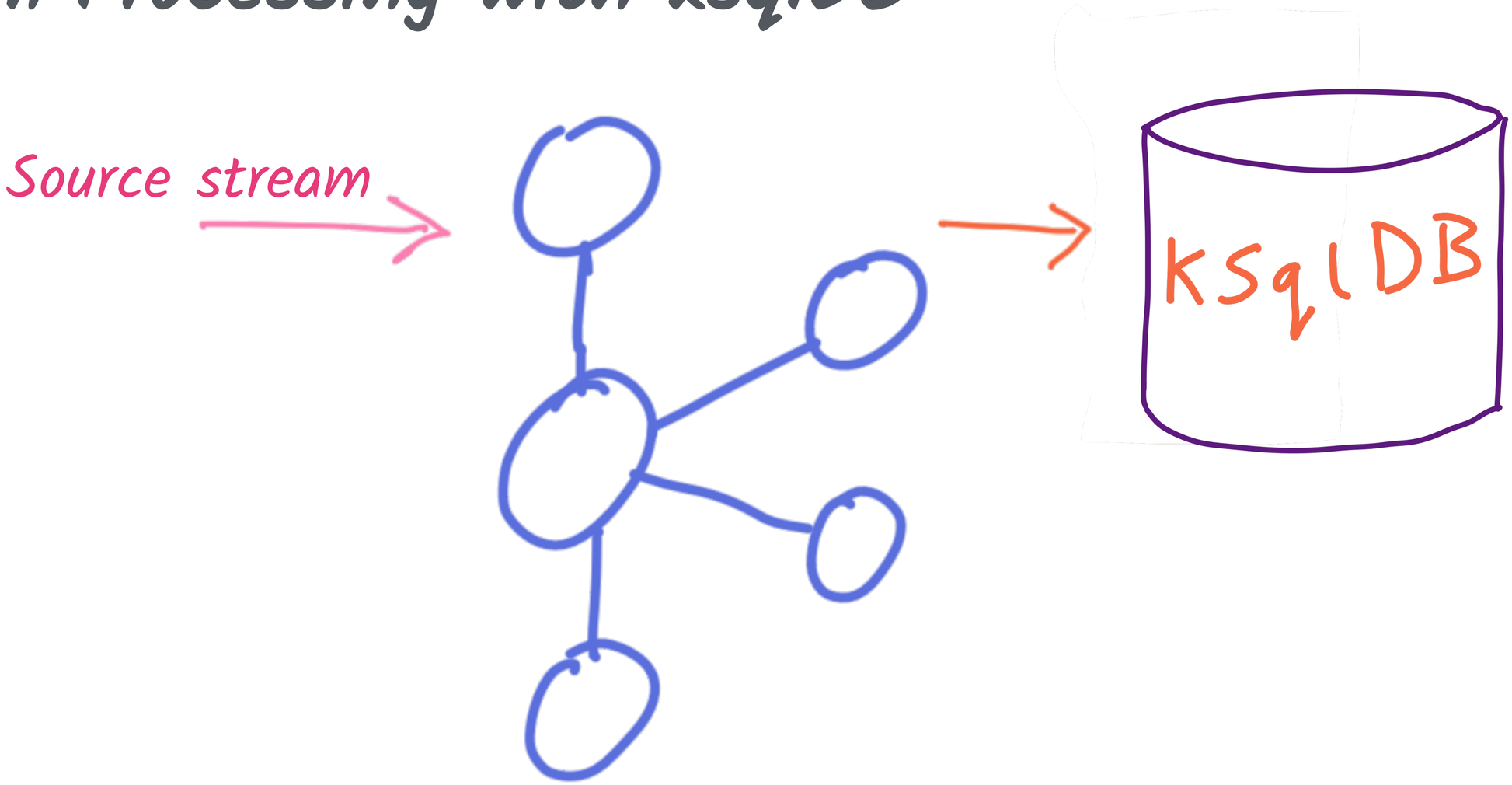

```
{
    "reading_ts": "2020-02-14T12:19:27Z",
    "sensor_id": "aa-101",
    "production_line": "w01",
    "widget_type": "acme94",
    "temp_celcius": 23,
    "widget_weight_g": 100
}
```


Photo by Franck V. on Unsplash

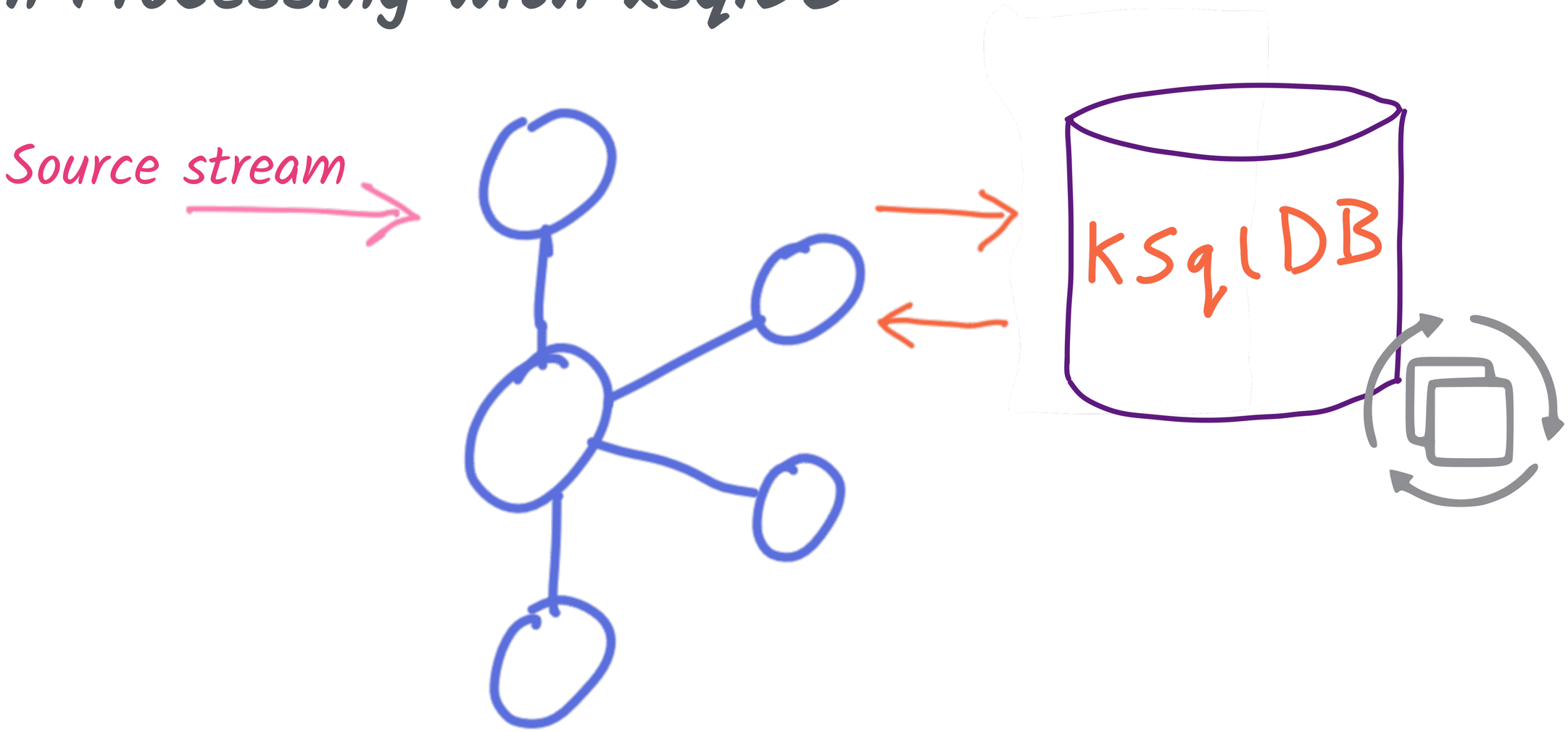
ksqlDB

The event streaming database purpose-built for stream processing applications.

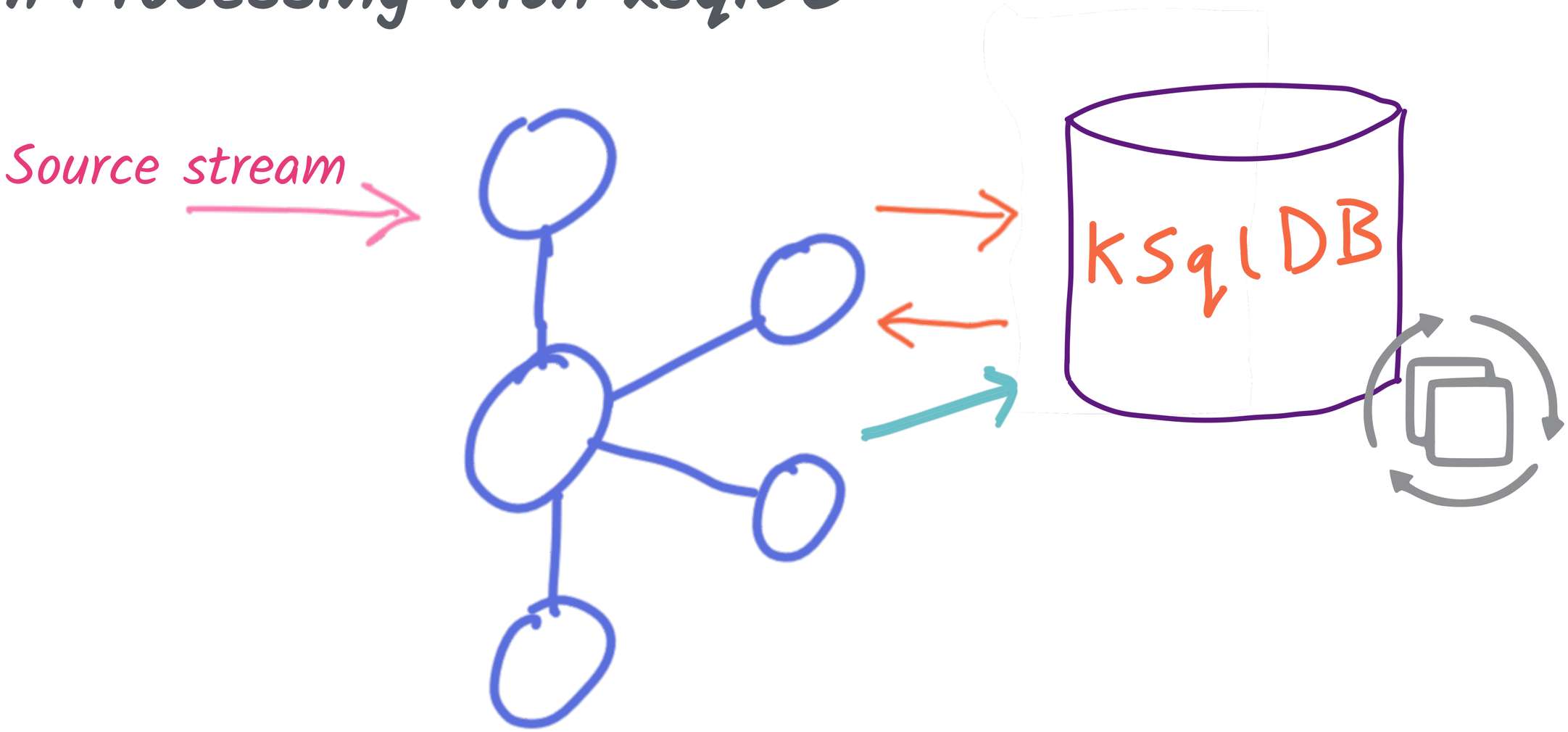
Stream Processing with ksqlDB



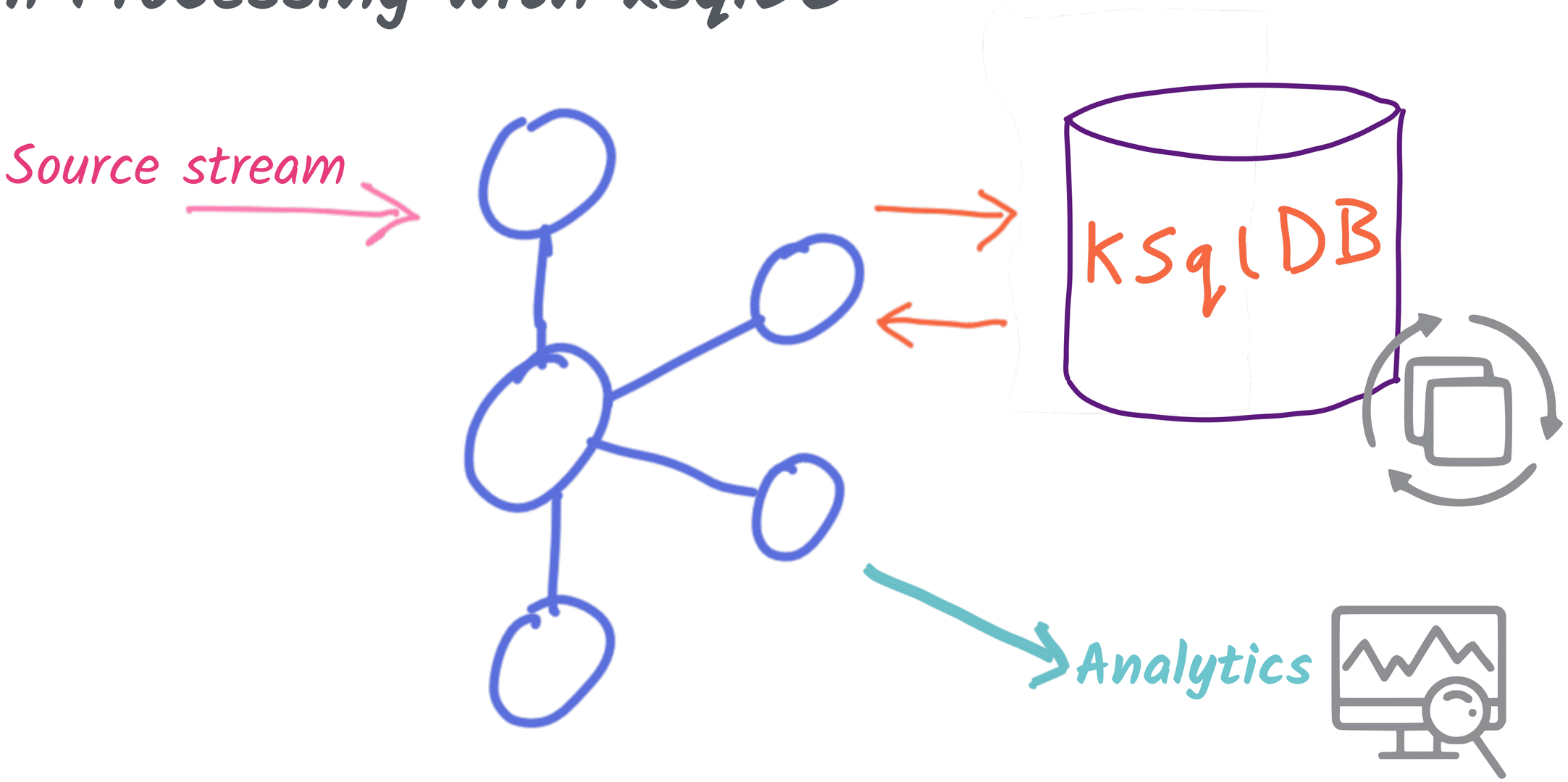
Stream Processing with ksqlDB



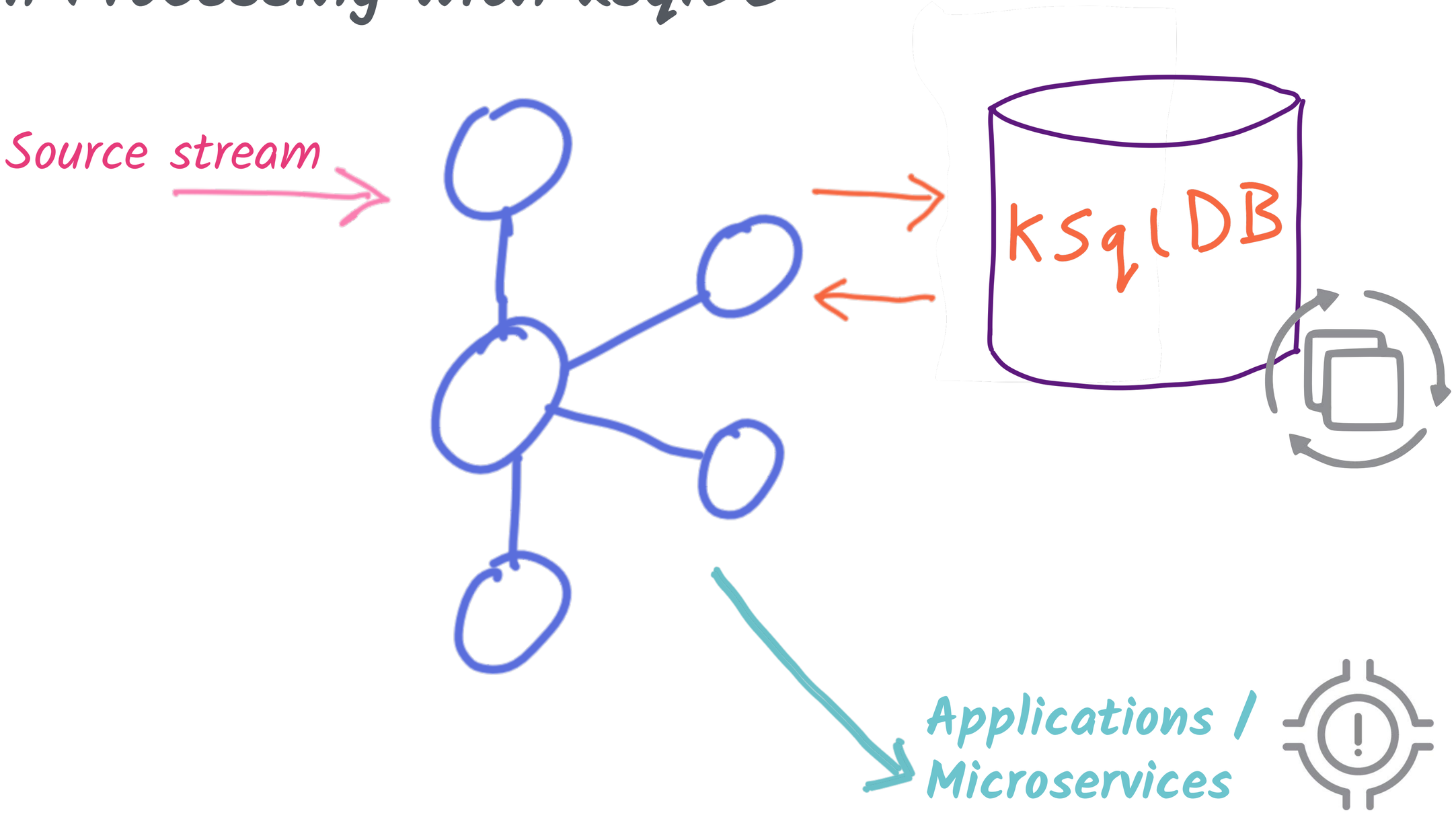
Stream Processing with ksqlDB



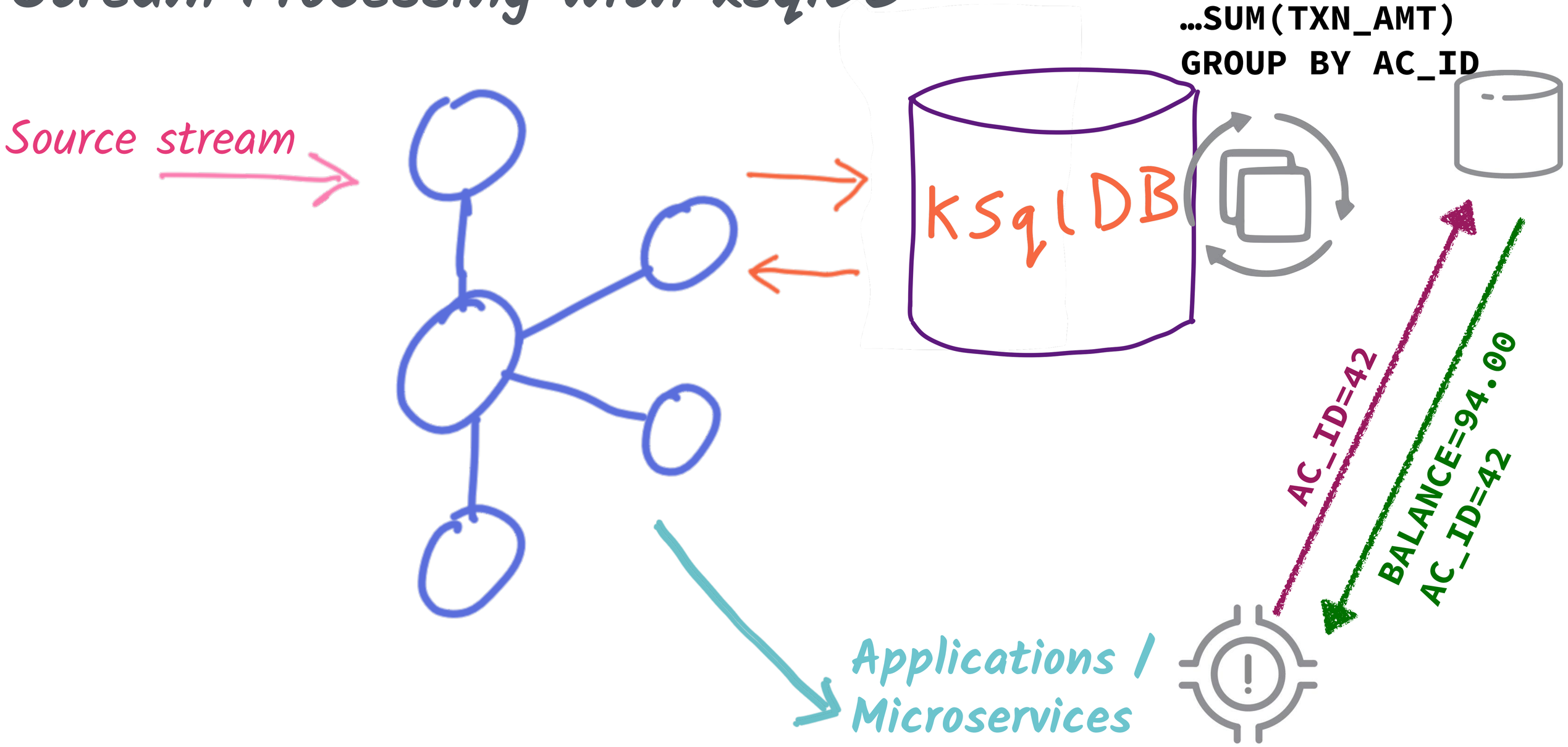
Stream Processing with ksqlDB



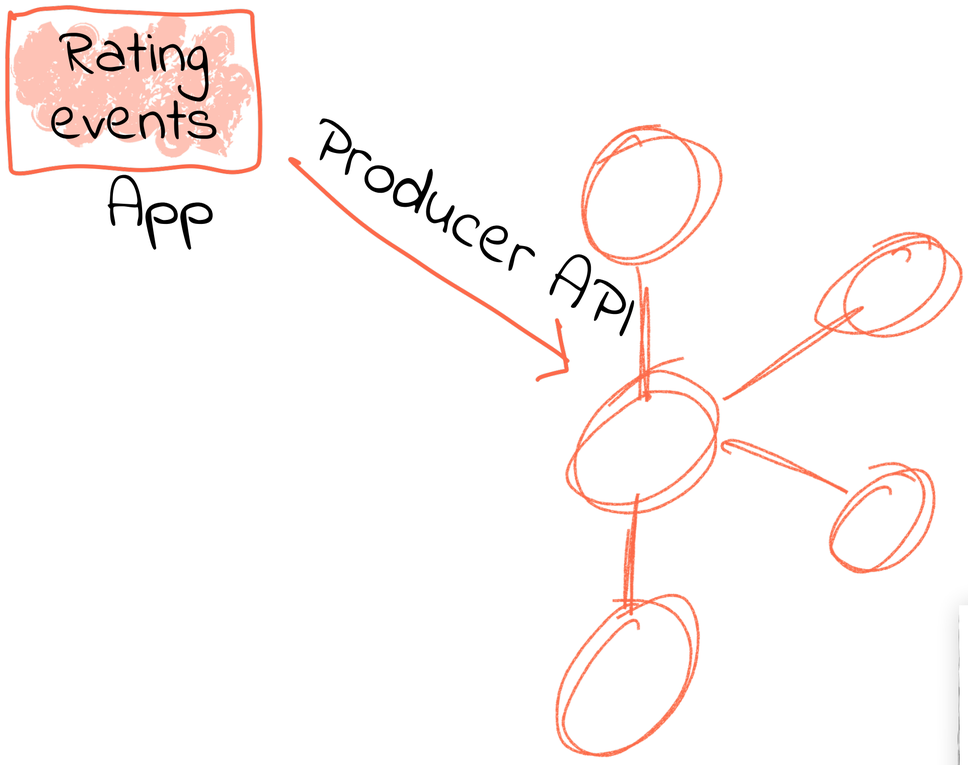
Stream Processing with ksqlDB



Stream Processing with ksqlDB

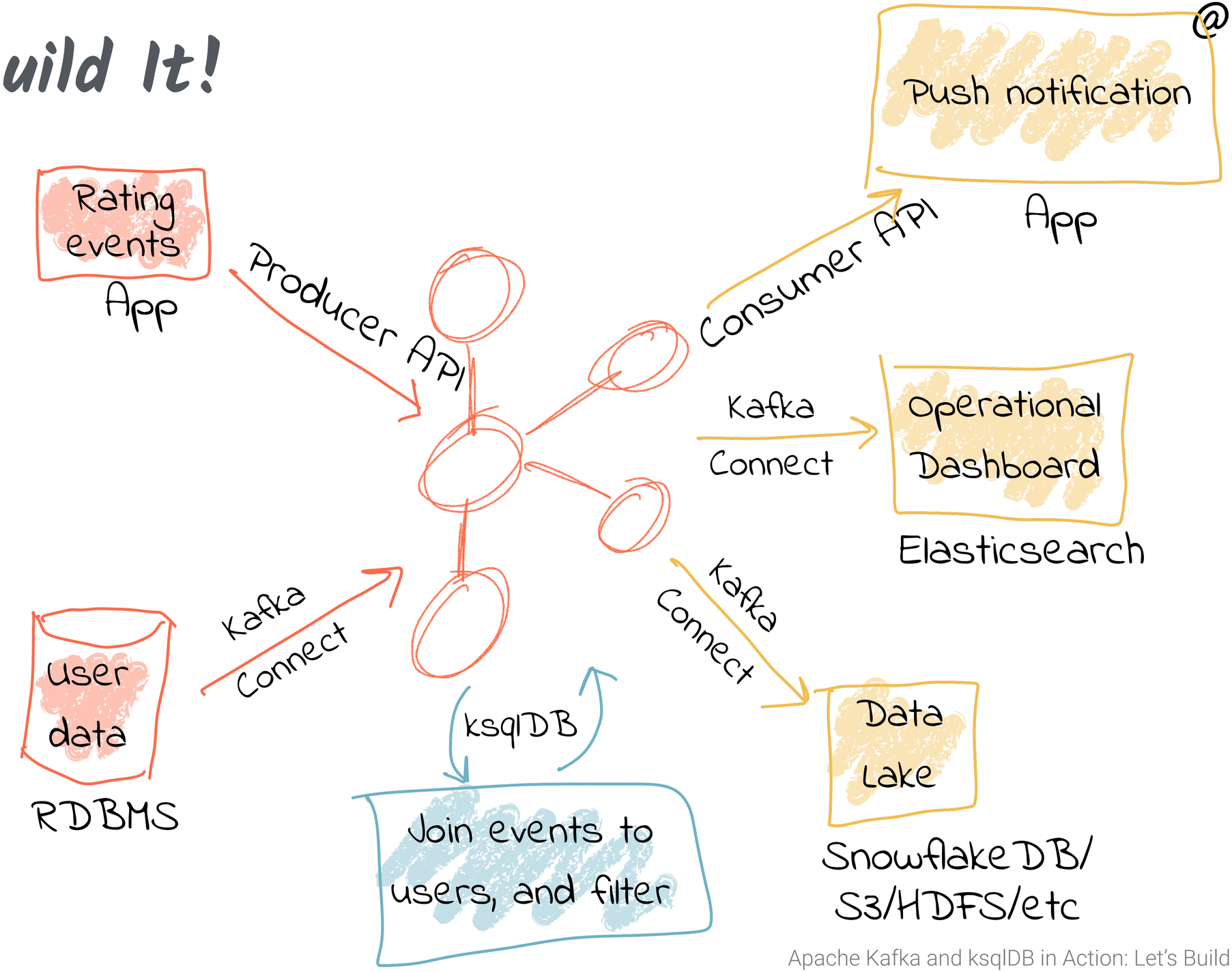


Let's Build It!

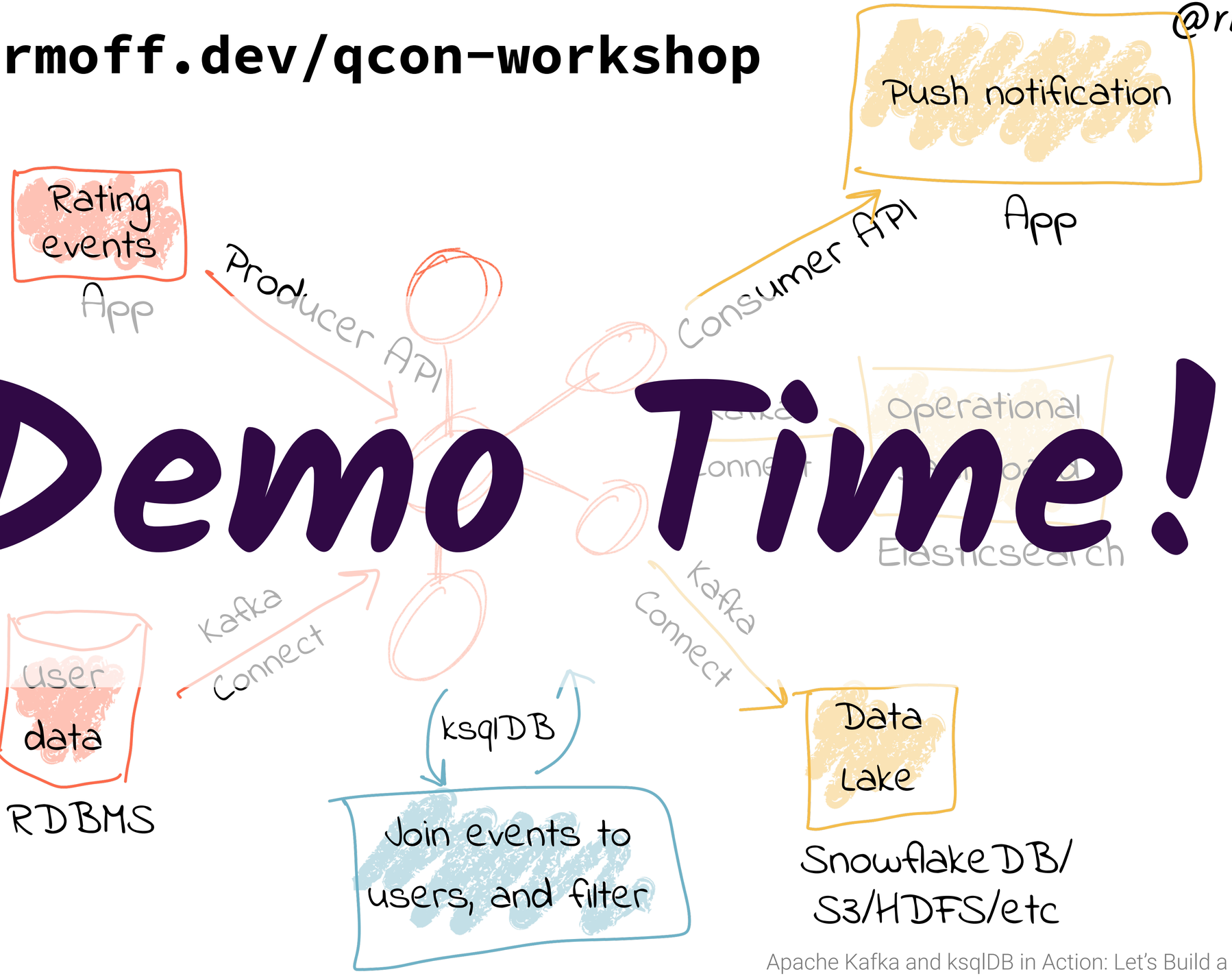



```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```

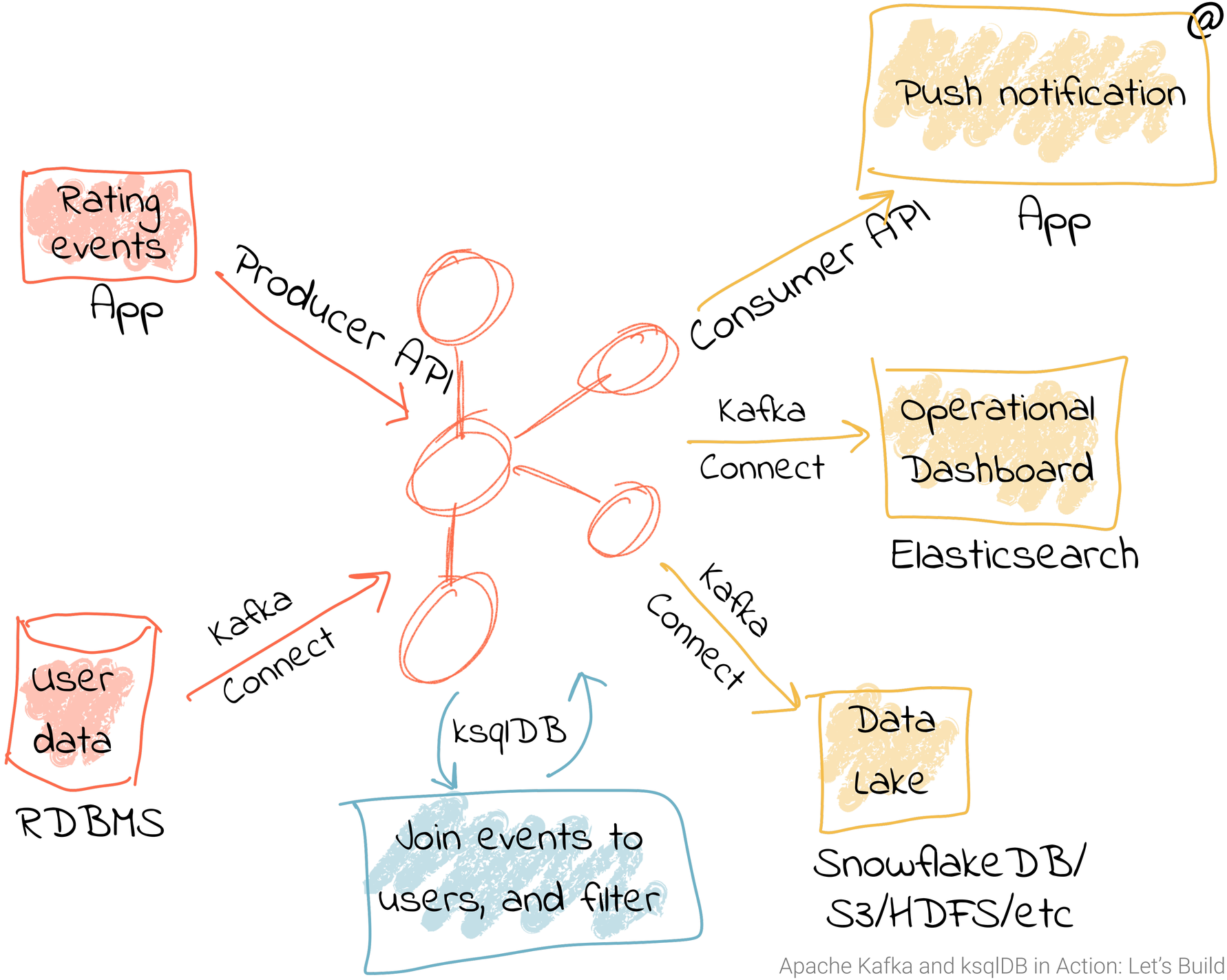
Let's Build It!

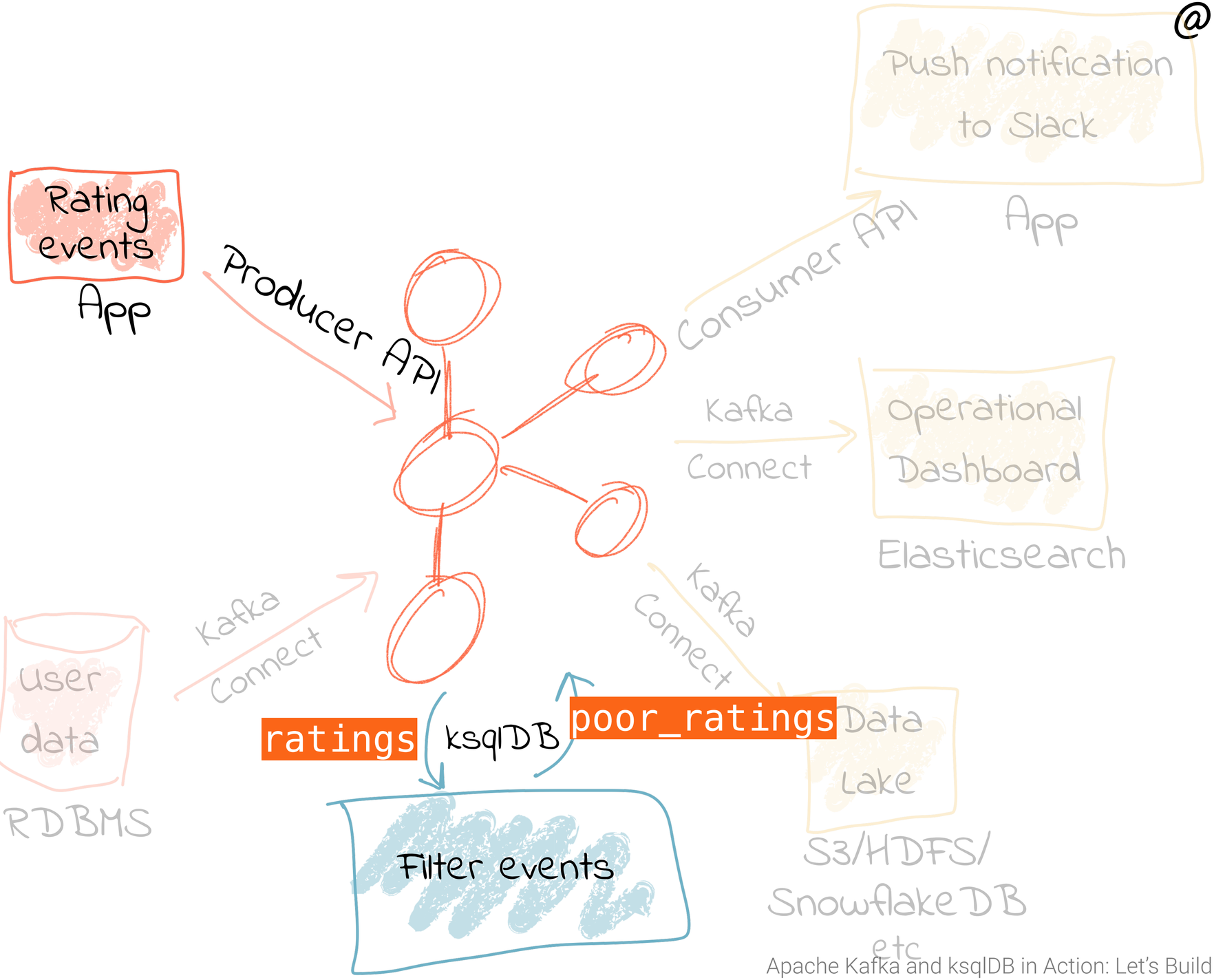


https://rmoff.dev/qcon-workshop



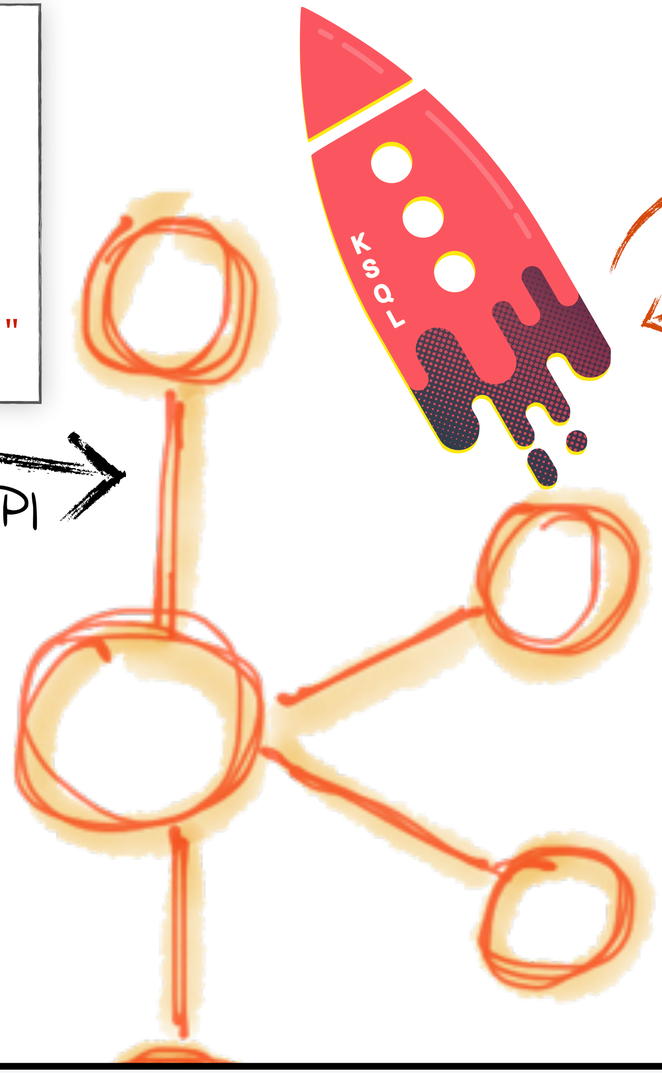
Demo Time!


Push notification
App
Rating events
App
Producer API
Consumer API
Kafka Connect
Operational Dashboard
Elasticsearch
user data
RDBMS
Kafka Connect
ksqlDB
Kafka Connect
Data Lake
SnowflakeDB/ S3/HDFS/etc
Join events to users, and filter


Rating events
App
Producer API
Push notification to Slack
App
Consumer API
Kafka Connect
Operational Dashboard
Elasticsearch
Kafka Connect
User data
RDBMS
Kafka Connect
ratings
ksqlDB
poor_ratings
Data Lake
S3/HDFS/ SnowflakeDB etc
Filter events

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```

Producer API

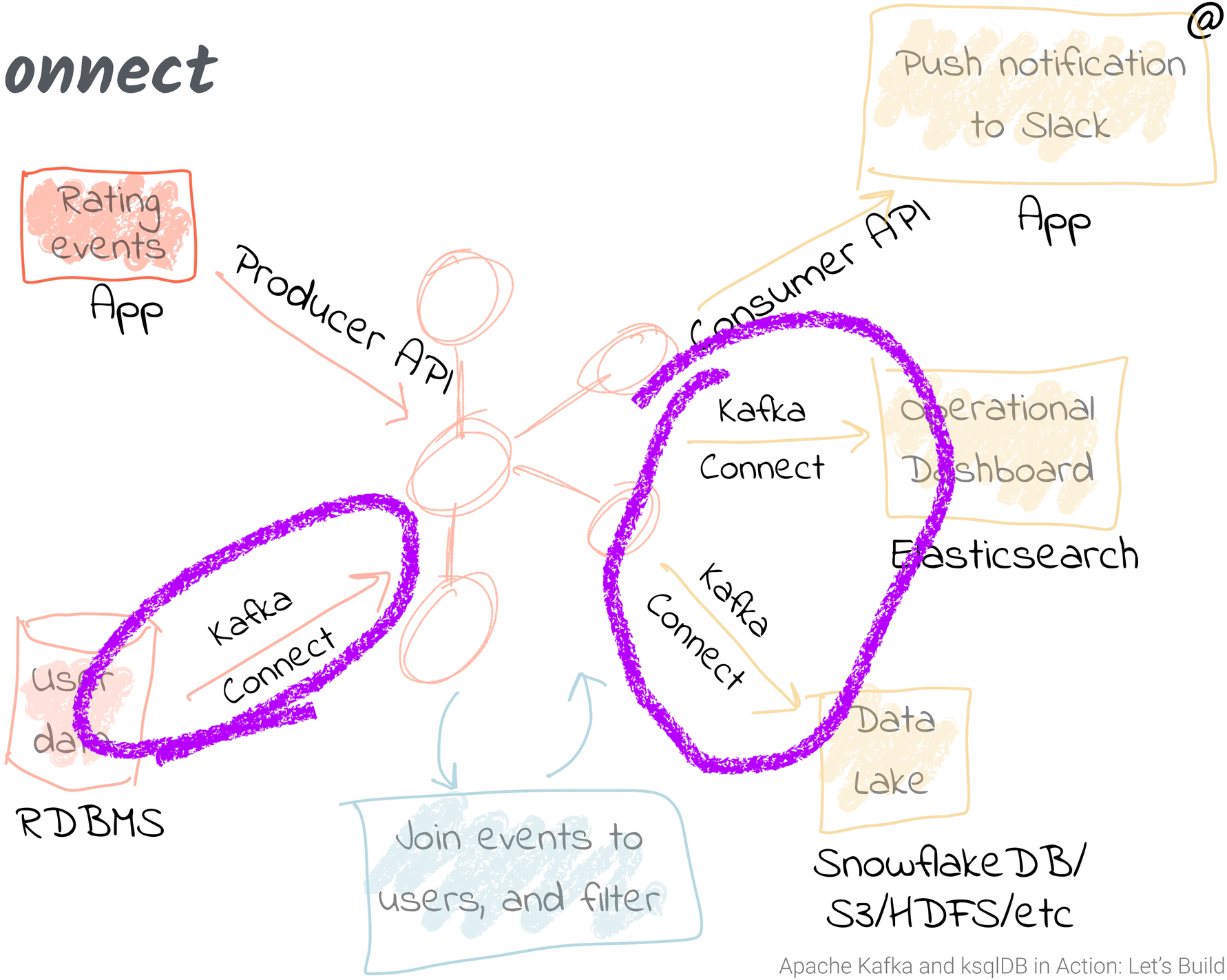


Filter all ratings where STARS<3

POOR_RATINGS

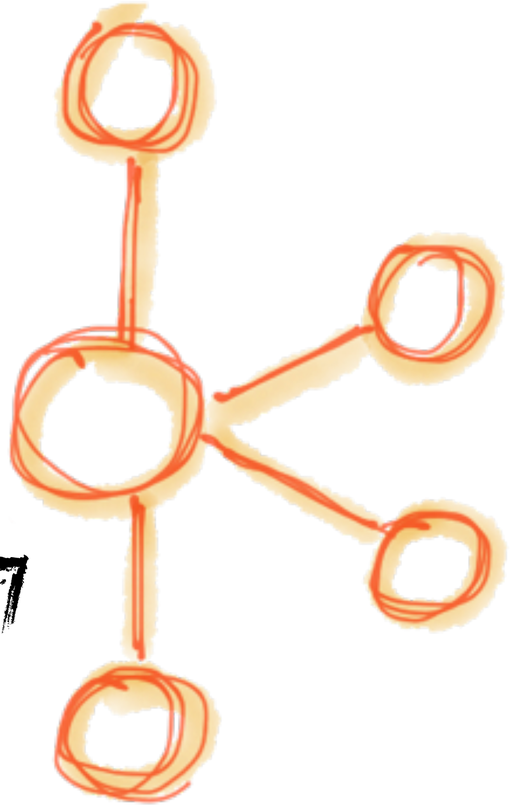
```
CREATE STREAM POOR_RATINGS AS
SELECT * FROM ratings WHERE STARS <3
```

Kafka Connect



```
{
  "rating_id": 2133,
  "user_id": 9,
  "stars": 1,
  "route_id": 7219,
  "rating_time": 1519402815063,
  "channel": "web",
  "message": "worst. flight. ever. #neveragain"
}
```


Producer API



Kafka Connect

Debezium

```
{
  "uid": 9,
  "name": "Neha",
  "locale": "en_US",
  "address_city": "Palo Alto",
  "elite": "P"
}
```

The Stream/Table Duality

Stream

Account ID	Amount
12345	+ £50
12345	+ £25
12345	- £60

Time

Table

Account ID	Balance
12345	£50

Account ID	Balance
12345	£75

Account ID	Balance
12345	£15

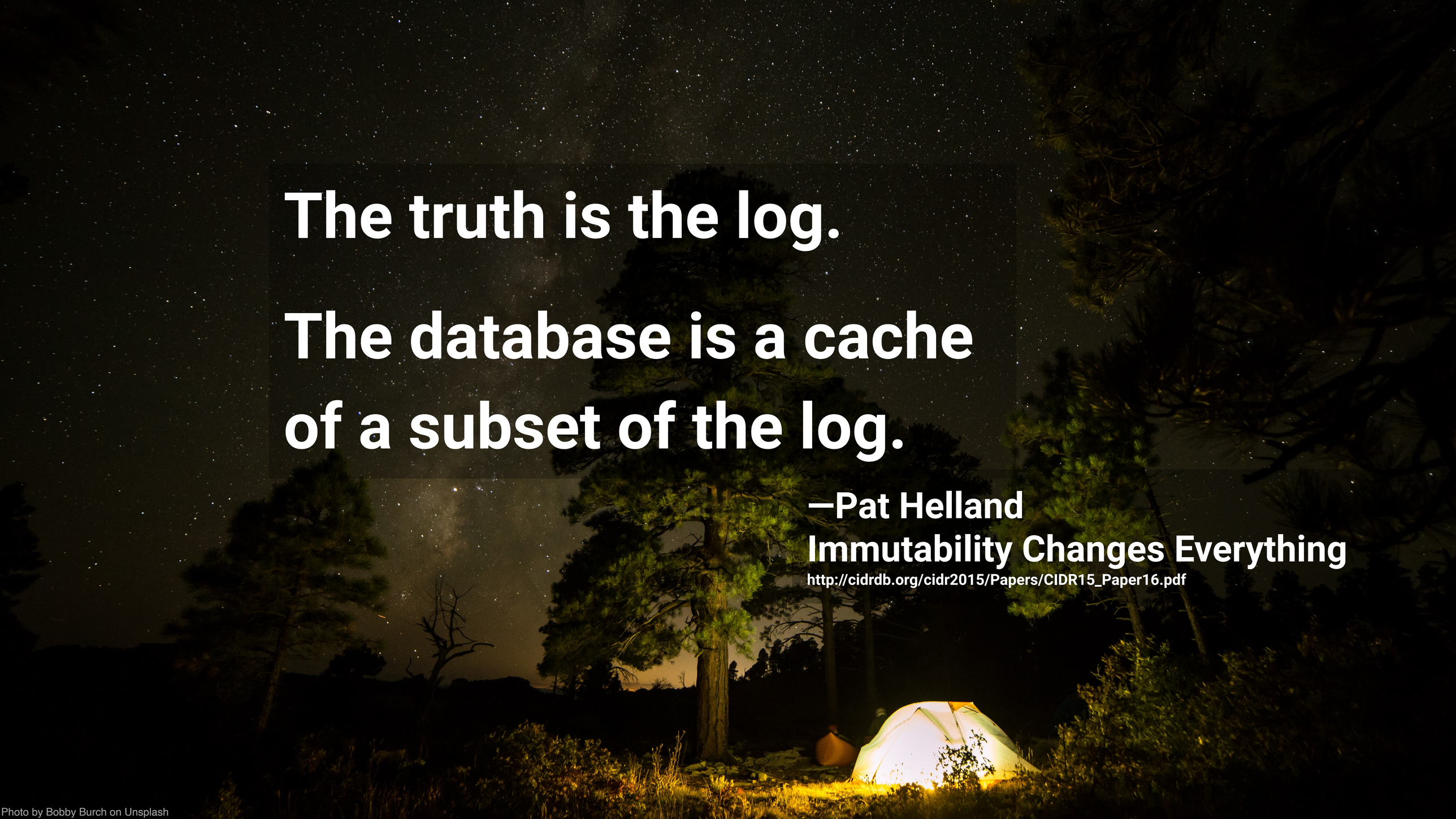
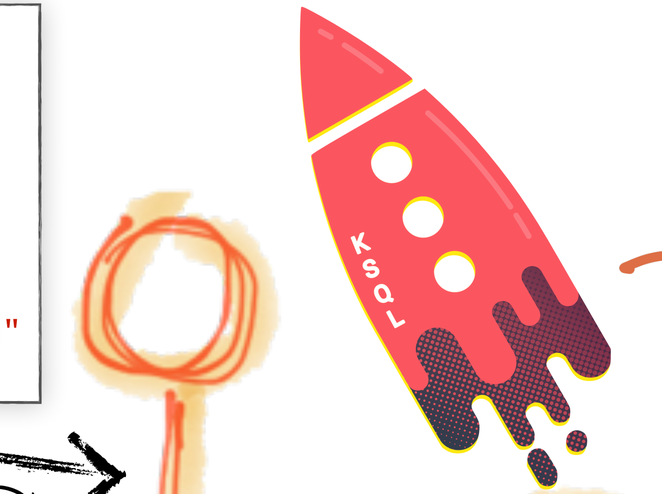


Photo by Bobby Burch on Unsplash

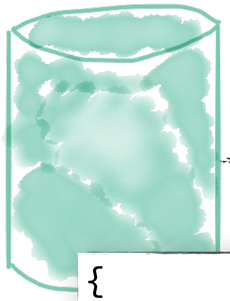
```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```


Producer API



Join each rating to customer data

RATINGS_WITH_CUSTOMER_DATA



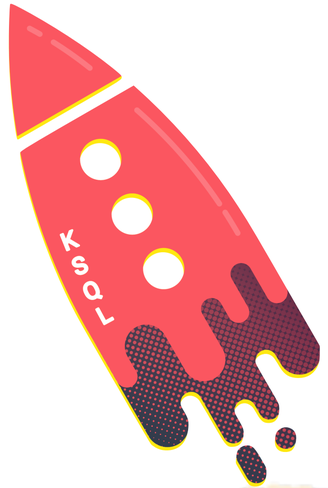
Kafka Connect


```
{
  "id": 3,
  "first_name
  "last_name"
  "email": "m
  "gender": "
  "club_statu
  "comments":
}
```


```
CREATE STREAM RATINGS_WITH_CUSTOMER_DATA AS
SELECT * FROM RATINGS
       LEFT JOIN CUSTOMERS ON R.ID=C.ID;
```

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```

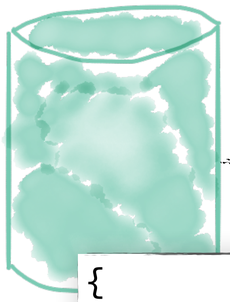

Producer API



Join each rating to customer data

RATINGS_WITH_CUSTOMER_DATA

Kafka Connect



Filter for just PLATINUM customers

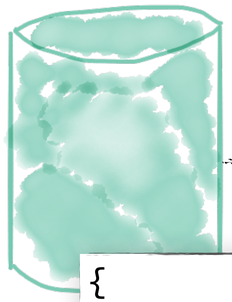
UNHAPPY_PLATINUM_CUSTOMERS


```
{
  "id": 3,
  "first_name
  "last_name"
  "email": "m
  "gender": "
  "club_statu
  "comments":
}
```


```
CREATE STREAM UNHAPPY_PLATINUM_CUSTOMERS AS
SELECT * FROM RATINGS_WITH_CUSTOMER_DATA
WHERE   STARS < 3
```

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304
 "channel": "web",
 "message": "worst. fli
}
```


```
CREATE TABLE RATINGS_BY_CLUB_STATUS AS
SELECT      CLUB_STATUS, COUNT(*)
FROM        RATINGS_WITH_CUSTOMER_DATA
            WINDOW TUMBLING (SIZE 1 MINUTES)
GROUP BY CLUB_STATUS;
```



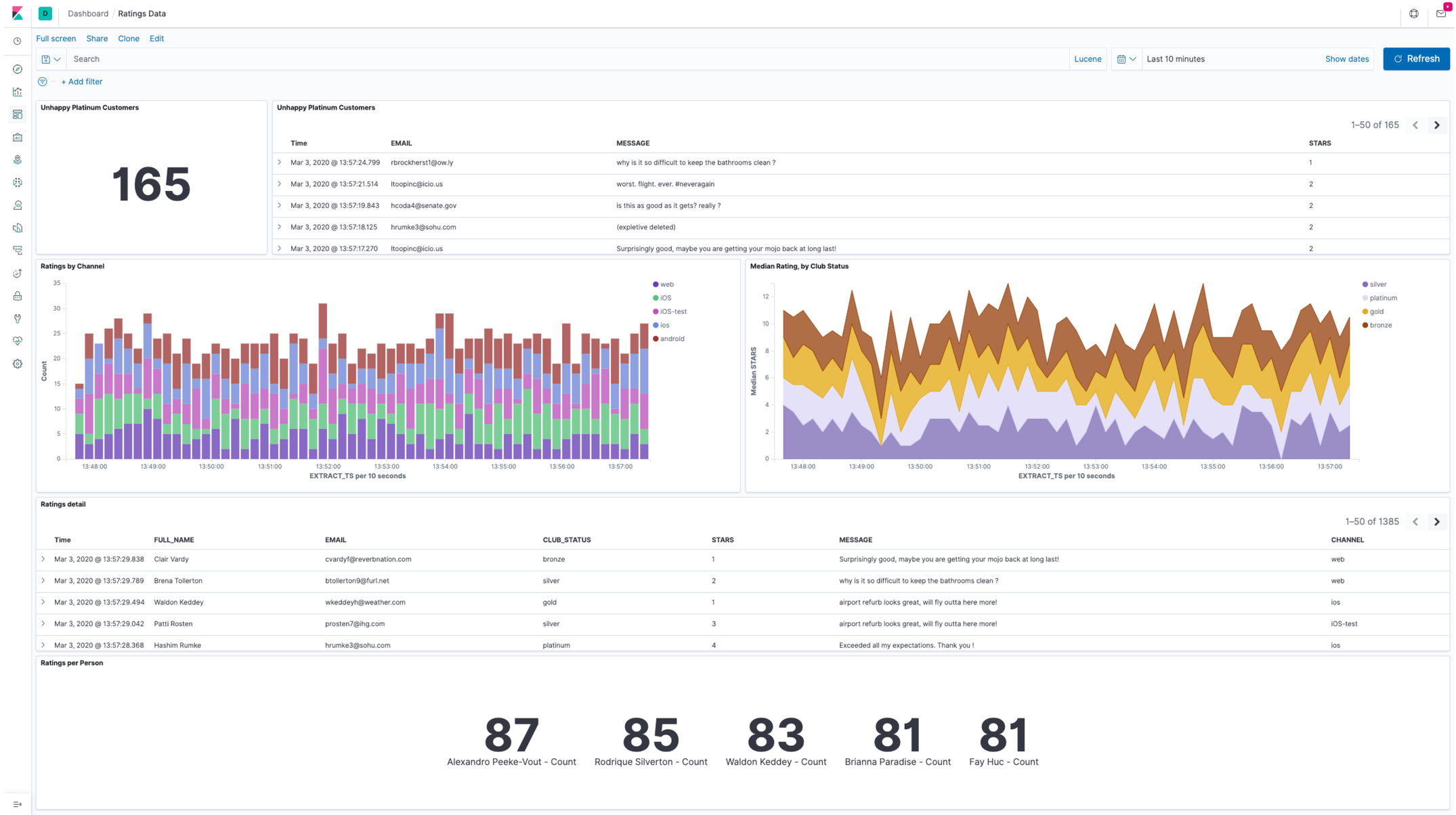
Kafka Connect

```
{
  "id": 3,
  "first_name": "Merilyn",
  "last_name": "Doughartie",
  "email": "mdoughartie1@dedecms.com",
  "gender": "Female",
  "club_status": "platinum",
  "comments": "none"
}
```


Aggregate per-minute by CLUB_STATUS

RATINGS_BY_CLUB_STATUS_1MIN

Kafka Connect → Elasticsearch



Want to learn more?

CTAs, not CATs
(sorry, not sorry)

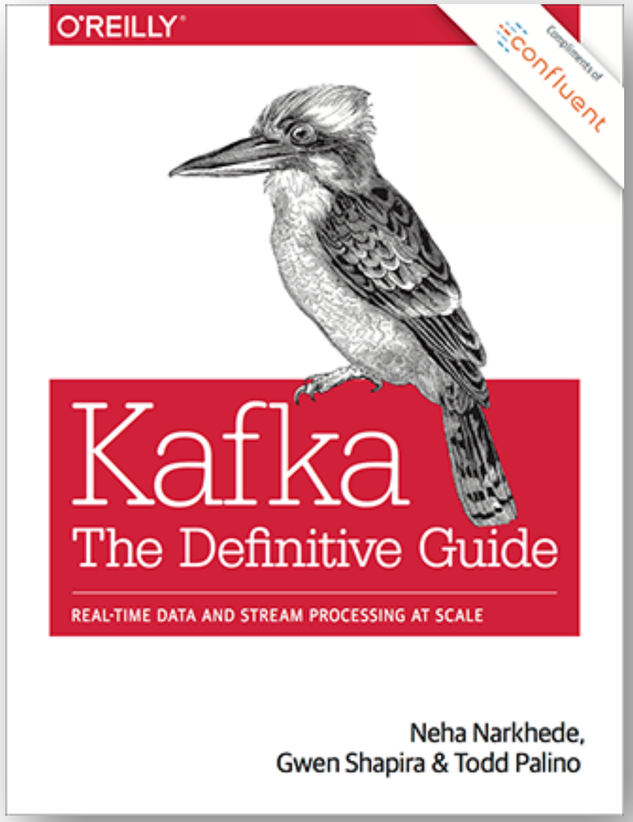
Learn Kafka.
Start building with
Apache Kafka at
Confluent Developer.
developer.confluent.io

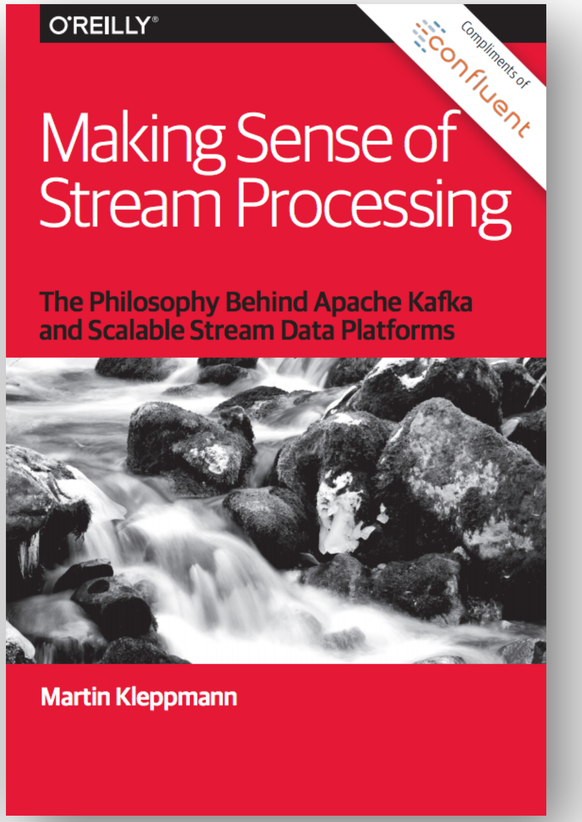
Confluent Community Slack group

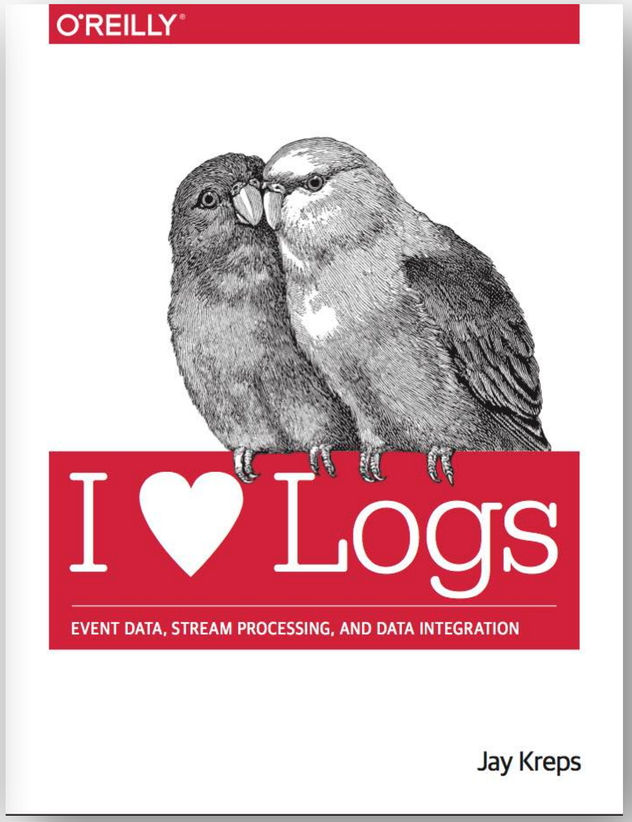
cnfl.io/slack

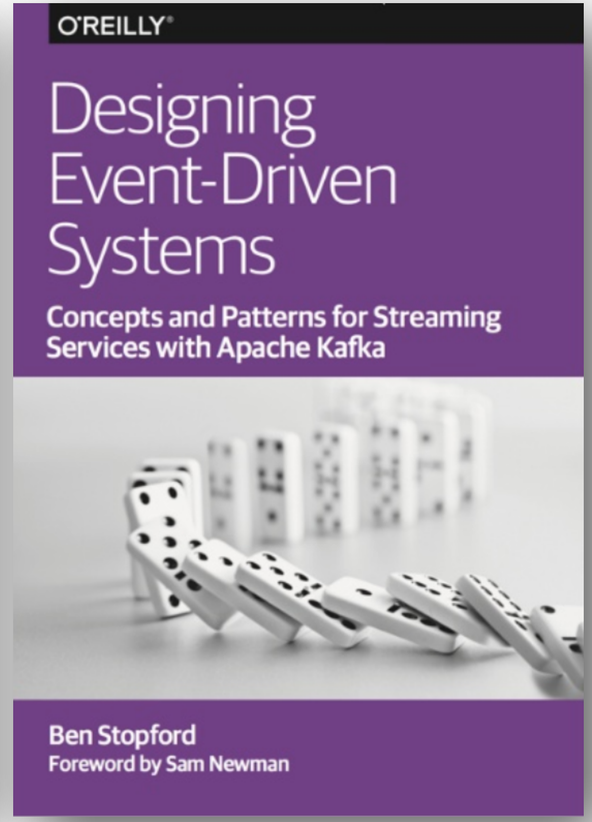
Free Books!

https://rmoff.dev/ljcjug









#EOF

Related Talks

- The Changing Face of ETL:
  Event-Driven Architectures for Data Engineers
  - 📖 Slides
  - 📽️ Recording

- ATM Fraud detection with Kafka and ksqlDB
  - 📖 Slides
  - 👾 Code
  - 📽️ Recording

- Embrace the Anarchy: Apache Kafka's Role in Modern Data Architectures
  - 📖 Slides
  - 📽️ Recording

- Apache Kafka and ksqlDB in Action : Let's Build a Streaming Data Pipeline!
  - 📖 Slides
  - 👾 Code
  - 📽️ Recording

- No More Silos: Integrating Databases and Apache Kafka
  - 📖 Slides
  - 👾 Code (MySQL)
  - 👾 Code (Oracle)
  - 📽️ Recording